Design • Build • Program • Learn • Compete

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

February 2012

Replace Humans In Hostile Environments With The Pathfinder Rover

✦ TEARDOWN
A Peek Inside the NEATO XV-11 Home Vacuum Cleaner Robot

✦ Eye Surgery Robot

✦ Parallax's Eddie

✦ Smart Shopping Cart

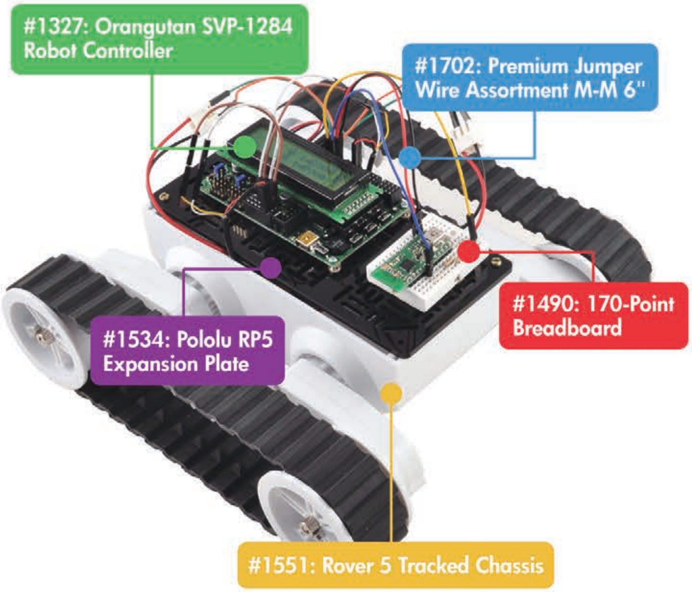
U.S. $5.50 CANADA $7.00

02

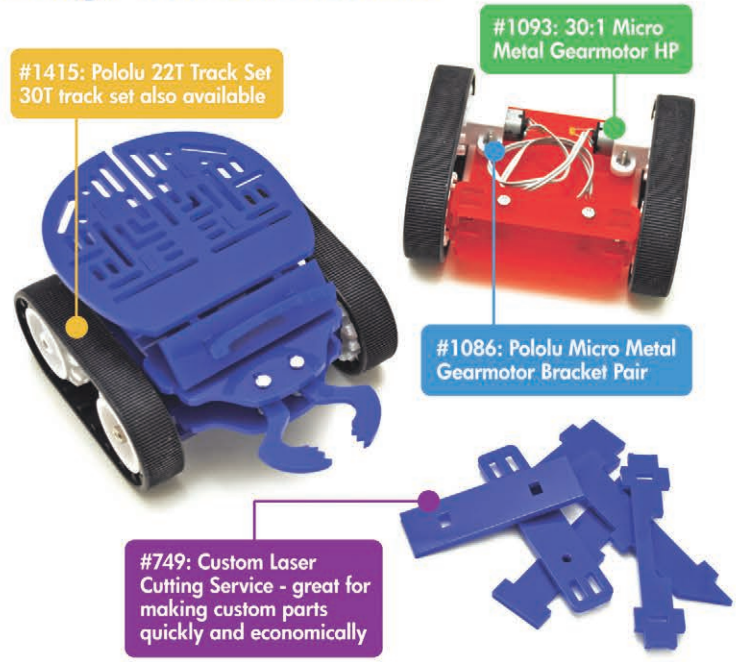
0 71486 02422 4


LISTEN UP!
YOUR BOT IS TALKING!
Knowing what your robot is doing can make a world of difference when you are competing. Our HTS-SS Blue Sensor station and series of sensors puts valuable GPS, speed, altitude, RPM, temperature, voltage and current information at your fingertips. Couple our HTS-Voice with your Hitec radio and your bot will actually tell you the data you want to hear. Plug the HTS-Navi into your PC and analyze your data at the end of the day!
HITEC 2.4GHz
ADAPTIVE FREQUENCY HOPPING SPREAD SPECTRUM
GET EQUIPPED!
GET EMPOWERED!
GET HITEC!
HTS-VOICE
HTS-SS BLUE
HTS-NAVI
[Current Sensor]
Current : 95.2A / C200
Wattage : 1332.8W
Warning (Hi) : 100A
3/3
MAX 98.0A
MAX 1421W
RST
AURORA 9
AURORA 9
Installed with the
HTS-VOICE
HITEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrcd.com

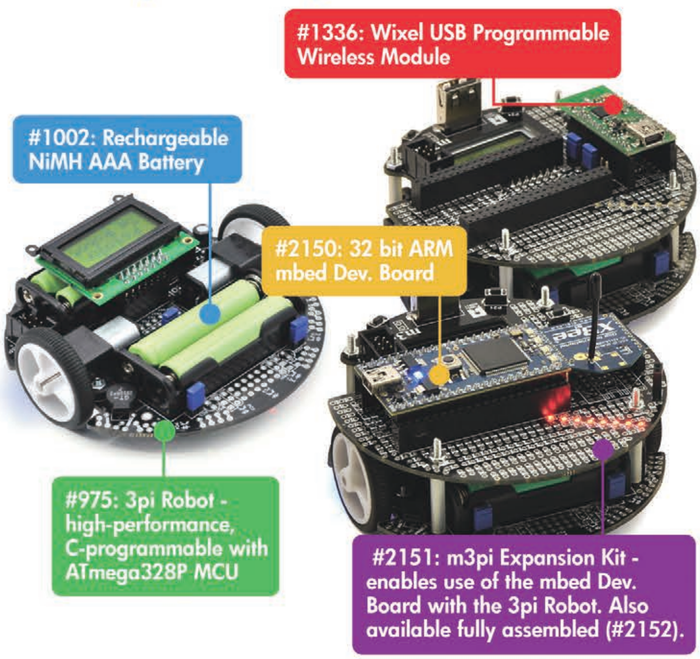
DIY Projects:
Simple Tracked Robot



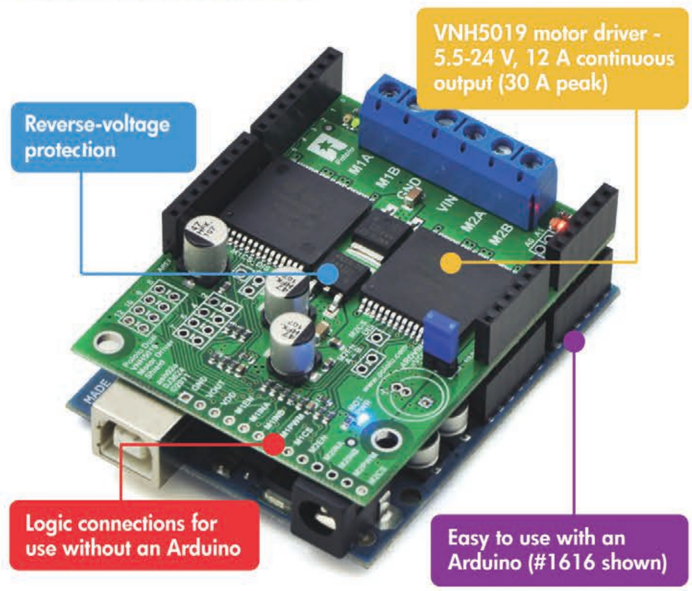
Custom Laser Cutting:
Design Your Own Chassis



Robots and Robot Kits:
Pololu 3pi and m3pi



Motor Drivers:
Item #2502: Dual VNH5019 Motor Driver Shield for Arduino



Finding the right parts for your robot can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products - from actuators to wireless modules - that can help you take your robot from idea to reality.

Pololu
Engage Your Brain

Find these products and more at **www.pololu.com**

SERVO MAGAZINE

02.2012
VOL. 10 NO. 2

PAGE 74

The Combat Zone...

Features

Events

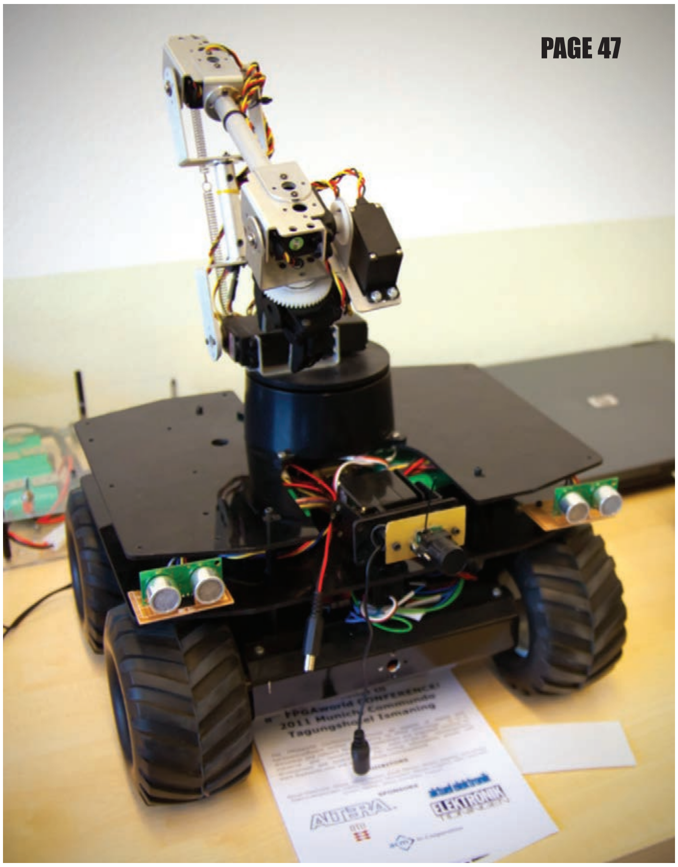
Columns

Departments

PAGE 20

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...



PAGE 47

PAGE 52

PAGE 40

Due to unexpected delays in acquiring necessary parts for his build, Matt Bunting's next installment in his series on "Pimp My Hexapod" will not appear in this issue, but will return as soon as possible.

Mind / Iron

by Bryan Bergeron, Editor

Behavior Change — A Hallmark of Intelligence

Intelligence — broadly defined — is the ability to adapt. In nature, the simplest life forms adapt through swapping genetic material during reproduction and through genetic mutation. For example, if one bacterium out of 5,000,000 carries a mutation that enables it to detect and avoid a poisonous chemical, then the descendants of that bacterium may possess the same ability. Mutations allow for changes in behavior at the population level.

Humans and other higher life forms also adapt through swapping of genetic material and mutation, but a significant change in instinctive behavior through these mechanisms requires tens to thousands of years. In the short term, mice and men adapt through learning. As a result, no two humans – or mice – will respond in exactly the same way to changes in the environment. In the face of danger, one might stay and fight, one might run away, and one might freeze with fear. The best course of action depends on the circumstances.

A simple carpet crawler that is programmed to respond consistently to a given stimulus is hardly considered intelligent. Assuming you have a rover using the typical subsumption architecture in which overall behavior is based on layers of fixed stimulus-response code, the robot will behave predictably. Fixed responses can be a good thing for a robotic welding machine that must replicate the same weld hundreds of times a day. However, if the goal is to enable a carpet crawler to escape a maze in the shortest time possible, the ability to adapt is essential. Machines with intelligence can have practical uses, as well. Consider a toaster that learns to associate, say, your fingerprint or even your face with your toast preference.

The point of this editorial is to encourage you to start experimenting with various forms of learning. If you're new to machine intelligence, I suggest you start simple by adding some randomness to your robot's behavior. For example, when the bumper sensor on your carpet crawler is activated, does your bot always reverse to the left? If so, why not make its behavior a little more interesting by having it either back up to the right, left, or straight back, based on a randomly generated number. Next, try experimenting with different weights – say 60% of the time, have your crawler move back to the right, 20% of the time have it move straight back, and 20% of the time have it move back to the left. Once you have a feel for basic behavior combinations, start investigating the myriad approaches to machine intelligence. Again, before you start coding neural networks on field programmable gate arrays (FPGAs), go for simple. Instead of manually assigning weights to random behavior, enable your bot to learn through experience.

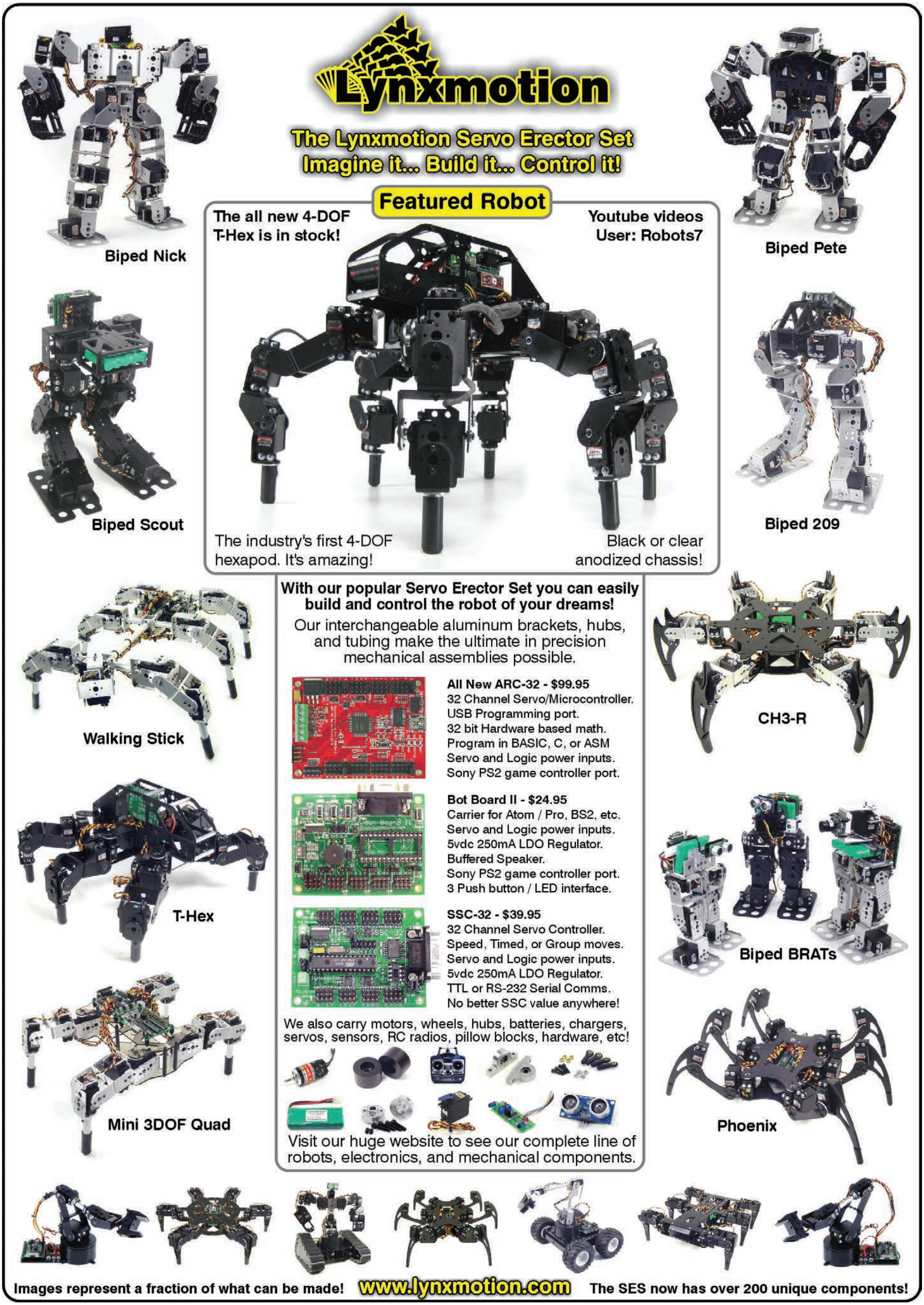
For example, let's say your carpet cawler upon colliding with a barrier, reverses to the left, right, or back at random. Statistically, your bot should exhibit one of these three behaviors one-third of the time. Now, let's create a rule that says if, during the execution of one of these behaviors that a rear bumper sensor is activated, then that behavior should be less likely. In other words, your program should automatically reduce the weight placed on that reversing behavior because it simply makes matters worse.

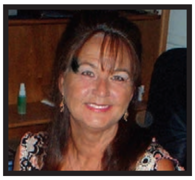
Whether it's determining how to reverse course or when to accelerate, the best behavior depends on the circumstances and the environment. I suggest that the goal isn't to simply make behavior change arbitrarily, but to enable your bot to adapt to the environment, That is, to learn. As you'll discover, fully exploring adaptive behavior may require an investment of a few more sensors and more processing power, but the journey is well worth it. **SV**

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The all new 4-DOF T-Hex is in stock!
Youtube videos User: Robots7
The industry's first 4-DOF hexapod. It's amazing!
Black or clear anodized chassis!
Biped Nick
Biped Pete
Biped Scout
Biped 209
Walking Stick
CH3-R
T-Hex
Biped BRATs
Mini 3DOF Quad
Phoenix
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible.
All New ARC-32 - $99.95
32 Channel Servo/Microcontroller.
USB Programming port.
32 bit Hardware based math.
Program in BASIC, C, or ASM
Servo and Logic power inputs.
Sony PS2 game controller port.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Visit our huge website to see our complete line of robots, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has over 200 unique components!

Robytes

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com.

by Jeff and Jenn Eckert

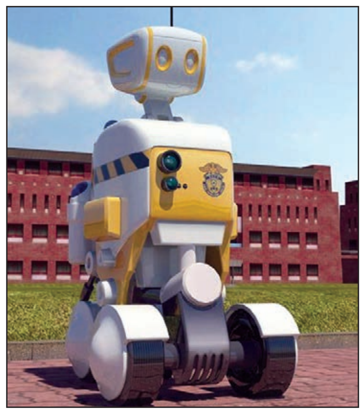
IHMC's FastRunner sprint bot.

Sprint Bot Mimics Ostrich

Pretty obviously patterned after an ostrich is FastRunner — an upcoming "dynamic bipedal platform capable of traversing moderately rough terrain as fast as the best human sprinters." Also obvious is its potential for military duties, so it's not surprising that the folks at the Defense Advanced Research Projects Agency (**www.darpa.mil**) are behind it. Doing the grunt work is a team of researchers from the Massachusetts Institute of Technology (MIT, **www.mit.edu**) and the Florida Institute for Human & Machine Cognition (IHMC, **www.ihmc.us**). The goal is first to demonstrate a realistic simulation of the bot running at 20 MPH on flat ground and 10 MPH on rough terrain, then to build a full planar FastRunner that actually accomplishes that. At present, the bot is reaching a simulated 20 MPH from a dead stop in less than 15 sec, and the running cycle is stable without control feedback. The hardware design is said to be 40 percent completed, and one full-scale leg has been machined via rapid prototyping techniques. Eventually, the sprint bot could achieve speeds as high as 50 MPH. Another of the team's objectives is to release FastRunner's open source simulation software; you can download it yourself at **www.ihmc.us/groups/fastrunner/wiki/1793c/FastRunner_Source_Code.html** (Mac OS is not currently supported, but Windows or Linux will do). To see him in action, just log onto **www.youtube.com/watch?v=wt_vR4Q-emI**. Disappointingly, it doesn't appear that FastRunner has the ability to bury its head in the sand or lay three-pound eggs.

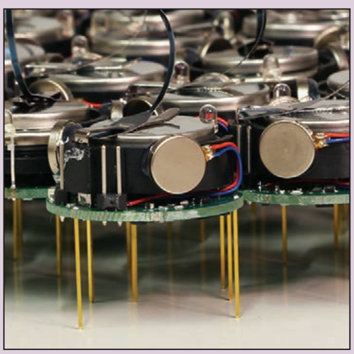
Bots Going to Prison

You may picture the typical prison guard as a uniformed thug with a permanent scowl, conspicuous bar fight scars, and a natural inclination to inflict pain. However, should you find yourself in a South Korean slammer, it may be a different story. The country's Ministry of Justice lately devoted $864,000 to the development of a team of robotic prison patrollers that make Asimo look like a fearsome beast. These "friendly robots" will perform some of the customary tasks, such as rolling through the corridors and watching for suspicious activities. But they are also designed to look after prisoners' well-being and even assess their mental states. Reportedly, even the smiley version shown here will undergo some last-minute cosmetic changes to make it look even more "humane and friendly." The first team is slated for duty in a penal institution in Pohang starting this March. How long the unarmed bots will survive among violent felons is anyone's guess, but the real dilemma revolves around what prisoners can use as a bribe to get cigarettes and crack.



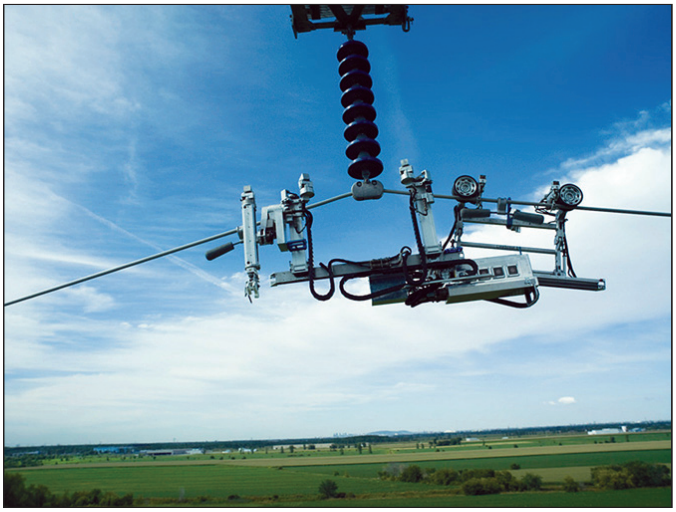
Prison guard bot will add another dimension to "soft on crime."

Swarms Now Available



A kilobot swarm may consist of hundreds or even thousands (hence, "kilo") of units operating in unison.

Last June, a Harvard Technical Report described the concept of "kilobots" which are little bug-like autonomous devices that scurry around on three legs. Twenty-five of the bots were shown to interact as a team, using coordinated behaviors such as synchronization, movement in formation, and mass foraging. Created by Harvard's Self-Organizing Systems Research Group (**www.eecs.harvard.edu/ssr/**), they were created to be inexpensive tools for educators, researchers, and anyone else who is interested in swarming behavior. The good news is that through a licensing agreement with Swiss bot builder K-Team Corporation (**www.k-team.com**), the little buggers will be on the market by the time you read this. All you have to do is log onto the RoadNarrows Robotics website (**www.roadnarrows.com**) and order all you want. The bad news is that "inexpensive" is a relative term. A ten-pack will run you $1,375, and you'll also need a controller ($615) and a charger ($226). But imagine how cool they'll look scrabbling around your kitchen floor. They might even scare away the cockroaches.



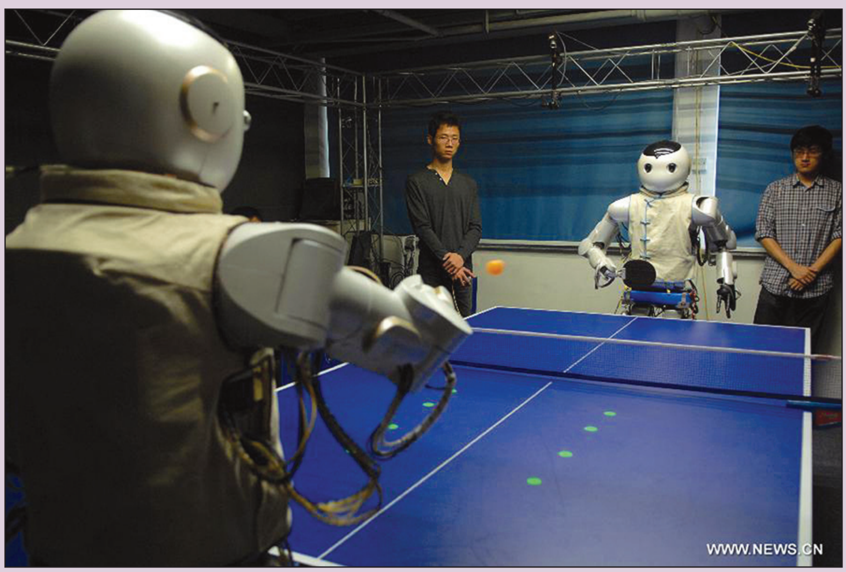
A LineScout inspection bot crossing an insulator string.

Looking at the Lines

When utility workers need to inspect transmission lines, they usually either move along the lines while dangling in special trolleys or peek at them from afar using binoculars. Neither method is exactly ideal. The former exposes them to as much as 735,000V of electricity, and the latter makes it easy to miss small but serious faults. However, Hydro Québec (**www.hydroquebec.com**) — a major provider of electricity to the US Northeast — has developed a robotic solution that goes by the name of LineScout. It presently uses five of them to maintain 18,000 miles of distribution lines. Its development required solving some interesting problems, including the need to withstand voltages that can easily fry electronic circuitry and interfere with radio control systems. It also has to withstand extreme weather conditions encountered in the northern areas of Canadian provinces. Perhaps the biggest challenge though was enabling it to maneuver past such obstacles as cable separation devices, porcelain insulators, corona rings, and the orange balls used to ward off low-flying aircraft. This is accomplished using pincers that allow it to lift its wheels off the powerline and swing them to the other side of an encountered obstacle. LineScout is actually controlled by a ground-based operator who makes use of its video cameras (four standard and one infrared) to detect hot spots and other defects. Via its robotic arm, the unit can also wrap and clamp frayed wires, measure resistance in splices, and even tighten bolts on equipment that is clamped to the wires. All of this adds up to a hefty load of equipment; LineScout weighs in at nearly 250 lb (112 kg). Nevertheless, it can scoot along at 1 m/s (3.3 ft/s) and operate up to five hours between battery charges.

Not Ready For The WTTC

When you talk about serious world-class table tennis, you're usually talking about the Chinese. It's therefore not particularly surprising that the first pair of serious ping pong playing robots has emerged from China's Zhejiang University (**www.zju.edu.cn/english**). "Wu" and "Kong" — two life-size humanoid robots — are programmed to serve, return, and keep score. They track the balls using eye-mounted cameras, predict their trajectory, and respond with the appropriate shot. The cameras — operating at 120 images per second — send images to the bots' processors which calculate the ball's position, speed, angle, path, and landing position in 50 to 100 ms, with an error margin of just 2.5 cm. The machines are capable of playing against humans as well, and one bot vs. human exchange involved crossing the net 144 times. They do have a long way to go before being ready to compete in the World Table Tennis Championship since they are not yet capable of advanced shots like curves, shanks, and slices. SV



"Wu" and "Kong" battle it out in ping pong.

GEERHEAD

by David Geer
Contact the author at geercom@windstream.net

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Eye Surgery Robot

To assist ophthalmologists who perform difficult eye surgeries, Thijs Meenink (a doctoral candidate as of this writing) and Dr. Ron Hendrix, PhD from Eindhoven University of Technology, the Netherlands (with input from Dr. Marc de Smet) have created an eye surgery robot. This robot can help surgeons to be more accurate, making new types of surgeries possible. The Eye-Rhas (Eye-robot-for-haptically-assisted surgery) system is ideal for retinal surgeries which require extreme precision in order to perform them.

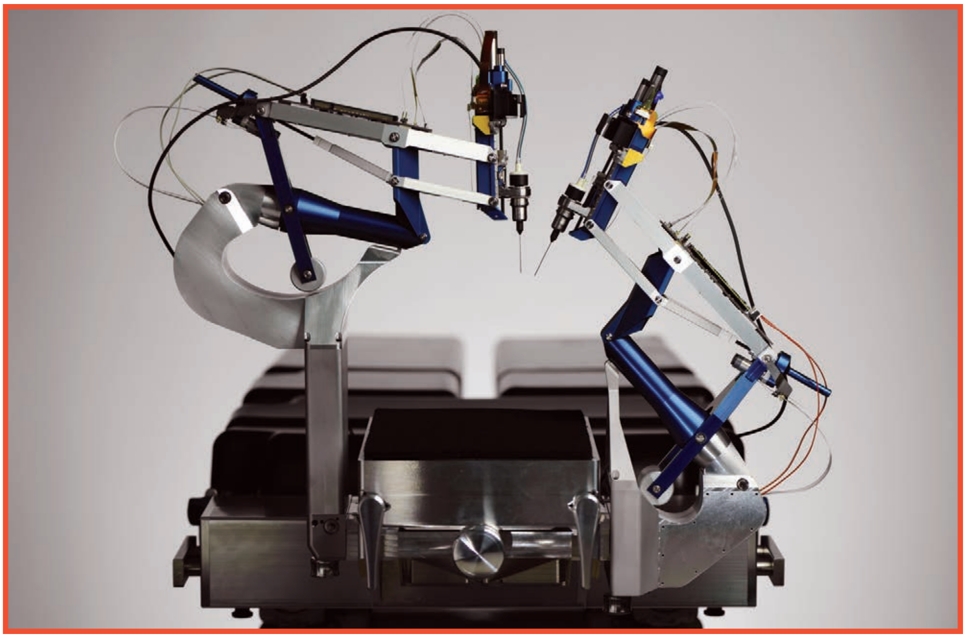
The Need

Ophthalmologists have made great strides with surgical achievements. "We used to take the whole lense out for cataract surgery. Now, we can place the lense in the eye. The end result is that the patient can see far away and read at the same time," says Dr. Marc de Smet. The surgeries are performed through increasingly smaller openings, offering quicker healing time and less trauma.

The tremor of any eye surgeon — young or old — is such that they can only operate at a scale of 80 to 100 microns in the best of circumstances, according to Dr. Smet. The eye surgery robot is expected to ease procedures at the 70 micron level while making them virtually error free. Using this robot, a surgeon could separate the retinal tissue flawlessly with very fine forceps.

The Eye-Rhas system will enable veteran surgeons who happen to have the most knowledge, experience, and skill to continue operating past the time when their hands trembling becomes worse.

"We first must go through a medical program to become ophthalmologists, then another program to become retinal surgeons. Then, we must acquire five years experience to work up to the most difficult procedures," explains Dr. Smet. By that time, a surgeon only has another 10 to 15 years of practice when they can operate successfully



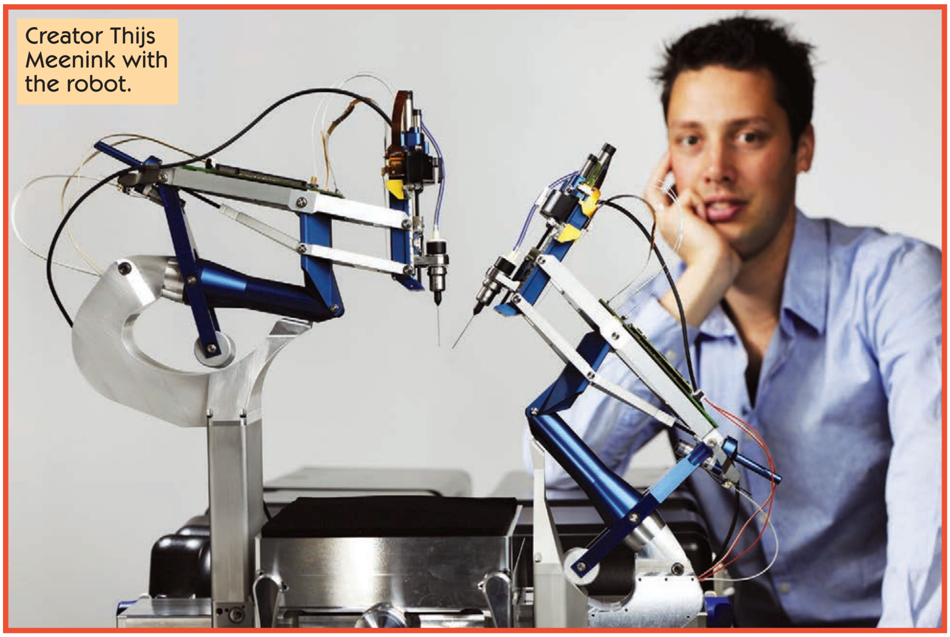
This photo shows the active part of the robot that performs the eye surgeries. The active part includes two instrument manipulators suspended via a short support arm on the right and a tall support arm on the left. The support arms permit pre-surgical adjustments in height and are connected to the headrest where the patient's head lies. Inside the headrest, the ophthalmologist can make X and Y adjustments in the horizontal plane. The surgical instruments consist of the partially black bodies with a hose at one end and the needle at the other end.

before hand tremors become an issue.

The Desire

According to Dr. Smet, younger surgeons like the use of robotics because it enables them to quickly become skilled at what they're doing. The robot uses haptic feedback to make it nearly impossible to cut the retina accidentally — which, of course, would be very bad. "The moment you touch the retina, you have damage," says Dr. Smet.

The robot can also shorten the time some surgeries take to finish. Some retinal surgeries take a long time because they require as many as 40 instrument changes. "When operating on a patient who has retinal detachments, the surgeon has to peel the membranes and remove membranes under the surface. These operations can last an hour or longer. A robot like this will make such difficult work more routine," comments Dr. Smet.


Creator Thijs Meenink with the robot.

An eye surgery robot could also enable the surgeon to simulate the procedure before performing the operation. Doctors could perform more sophisticated and intelligent surgeries and learn from simulations. "It will also be possible to operate in places in the eye where we can't today, perhaps underneath the retina," says Dr. Smet. The robot may also make it possible to bring in new layers of tissue using stem cells or generated tissues, explains Dr. Smet.

The robot is very mobile, weighing only 30 lbs, so it can be transported, set up, and used almost anywhere — even in the field such as in military deployments.

How It Works

The robot filters out the surgeon's hand tremors by enabling instrument control through a haptic interface (HI) which takes the form of a joystick. The forces of the instrument are measured by a nearby force/torque sensor. The system feeds these forces back via motors at the haptic interface to create torque. "This helps the surgeon's accuracy because he will have an additional perception where most surgical forces are below the detection limit," says Meenink.

Because the frame supports the HI, the physician only needs to operate it by hand, meaning that he/she does not have to support the weight of the HI while manipulating it. The surgeon's arms are supported on armrests, so he/she can attune their full focus and energy onto the surgery.

The control software also helps dissipate any hand tremors. Tremors are on one frequency while the intended movements are on another frequency. Because the hand movements of the surgeon are scaled down once they reach the instruments — which move at only a fraction of the measure of the hand movements — the system filters out what is left of the unintended movements. Specifically, the surgeon's movements are all scaled down mechanically. When the HI moves 18 cm, the surgical instrument's needle moves about 25 mm into the eye, according to Meenink.

The robot takes a number of other approaches to maintain surgical accuracy. For example, because the instruments the robot operates with are 0.5 millimeters in diameter (just like the ones used by surgeons by hand), an ophthalmologist can perform surgery without having to use sutures. "After removing the instrument for the last time, the wound is self healing. It does not require post surgical suturing," says Meenink.

The opening into the eye through which the instruments operate is a single opening. The needle enters the same hole each time to minimize the risk of permanent damage. Using four degrees of freedom, the instruments pivot at the opening to move where they need to go so that no further openings are required. "The mechanism that achieves this is kinematically defined," explains Meenink.

The robot is required to have a high level of stiffness, a low level of friction, and a play-free design. "These requirements contribute to a high positioning accuracy and safe usage," says Meenink. It was necessary to remove any free play from the design because it contributes to inaccuracies.

The robot also includes a mechanism that speeds up instrument changes, so ones that would normally take 15 to 40 seconds a piece take about five seconds each. To change instruments, the robot turns the instrument container using a rotary motor. A Z-drive — like a pushbutton pen — pushes the instrument out and opens a

clamp which can take the next instrument in and clamp it securely.

Ergonomics Make for Better Operations

In normal procedures, a surgeon has to look through a microscope. This means he has to work in a static and non-ergonomic body posture, notes Meenink. "We chose to implement a display for the visual feedback from the captured microscope images," says Meenink. (It is also possible to use the system with the microscope.) The display — like the armrest — reduces fatigue. The HI is also adjustable to deal with diversity in body dimensions and to guarantee an ergonomic working method.

The HI itself simulates the movement of the instrument tip inside the eye. A surgeon would normally have to move the instrument one way outside the eye for it to move the other way inside the eye. Now, the surgeon can move the HI in the same way as if he had the very tip of the instrument in his hand inside the eye.

There are also positioning concerns for the robot and the patient. The slave robot must be adjusted to each patient's head size. "There are pre-surgical adjustments for this," says Meenink. These adjustments are made and then locked in throughout the surgery. The robot must be positioned, as well. Because it is designed for vitreo-retinal eye surgery (surgery to the inner side of the eye, the vitreous humor and the retina) and because this cavity is only reached through a certain area of the sclera, the robot must be positioned to enter the eye there, according to Meenink.

Conclusion

Future plans for the eye surgery robot include optimizing its performance in accuracy, force feedback capabilities, and usability. Its creators are testing the robot to see just what needs optimizing, according to Meenink. Meenink believes the first use of the Eye-Rhas system on a human will occur within five years. In the meantime, he is looking into commercializing the technology. SV

Resources

The University of Technology
www.tue.nl

Thijs Meenink's page
http://w3.wtb.tue.nl/nl/people_pages/?script=showemp.php&pid=7524&all=1

R. Hendrix, Robotically Assisted Eye Surgery: A Haptic Master Console, PhD thesis, Technische Universiteit Eindhoven, 2011, ISBN 9789038624426.

H.C.M. Meenink, Vitreo-retinal Eye Surgery Robot: Sustainable Precision, PhD thesis, Technische Universiteit Eindhoven, 2011, ISBN 9789038628004.

A description of rotary encoders
http://en.wikipedia.org/wiki/Rotary_encoder

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by Dennis Clark

Another month, and more interesting questions have come my way. I will skip the traditional New Years Resolutions, because, well, it's February and I forgot to make them last month. I do resolve to make a totally new robot this year, so we'll see how that works out. This is a quiet time in the robot room — no imminent competitions, except that SparkFun will be having another AUV competition in a couple of months! Ahhhh!

Ahem ... let's move on and get to the questions for this month.

Q. I am Vikram from India. I am a higher secondary school student. I have been reading *SERVO Magazine* right from 2004 and I am a great fan of your column. I have built many robots to follow lines with PID control. For the last six months, I have been trying to build a Micromouse robot for the Singapore Robot Games' Micromouse JC and open. I built a robot using Pololu 50:1 micro metal gear motors, a Pololu wheel and encoder set, and three Sharp GP2D120 IR range sensors. The Sharp sensors are really reliable and I was able to implement PID control successfully. However, the response time of these sensors is about 40 milliseconds. My robot is really fast and it travels about 4-5 cms in that time. So, I was not able to make turns easily and my robot crashed in the initial search run itself. I was informed that Pico Mouse will be conducted instead of Micromouse from the next year in the Singapore Robotic Games.

Now, I finally come to my question: Which is the best analog wall sensor for Micromouse and Pico Mouse?

1. Sharp sensors are reliable, immune to external light and the reflectivity of the object, but they have slow response time, have a 4 cm minimum range, and are big in size — too large for Pico Mouse. Is there any way that I can hack these sensors to respond faster?

2. Ordinary IR sensors (Tx and Rx) are affected by wall color, reflectivity, and external disturbances. (I have painted my walls with white paint and there is a slight difference in reflectivity of each wall.) How can I make better, more reliable IR sensors?

3. TV remote sensors like the TSOP 1738 are immune to external light, but they give only digital output. Is there any way of making these sensors give analog proportional distance information for PID control? I tried varying distance sensitivity by changing frequency from 38-45 kHz to produce an analog output, but TSOP seems to receive a signal almost all the time. I have checked the signal using an oscilloscope and the signal seems to be very noisy.

4. Is there any way to create IR sensors based on signal triangulation, similar to Sharp sensors?

5. Can I use ultrasonic sensors on three sides to detect distance? (I have never tried this before.) Commercially available ultrasonic sensors like MaxBotix have long minimum range values and I heard that the ultrasonic waves are dispersed by the poles used to hold the walls and produce wrong distance readings.

6. Can I create my own ultrasonic sensors by using the ultrasonic transducers available from Jameco?

7. I have never tried using overhead IR sensors to sense walls. Is this easier and more convenient than using distance sensors?

8. Are there any books that I can refer to for Micro/Pico mouse?

9. Is it really essential to use analog sensors and PID control? Can I use multiple digital sensors instead and still use PID? Or, should I sacrifice speed and PID control for reliability?

Could you please help me with these problems which I have been trying to solve for the last six months? Your answer will really help me to compete in the Micromouse event in IITs and the Pico Mouse event in SRG.

— B. Vikram

A. Oh my gosh, Vikram! That is quite the question! I'll try to answer all of your questions, so bear with me. I'll answer by number. That way, I can keep myself

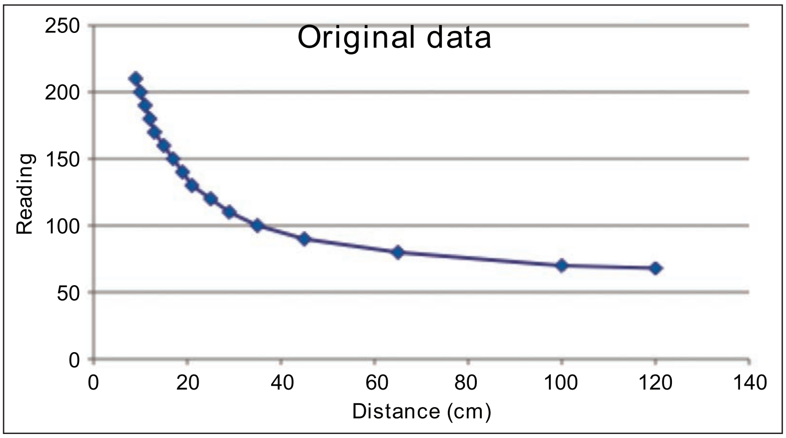


Figure 1. Sharp GP2D12 voltage vs. range.

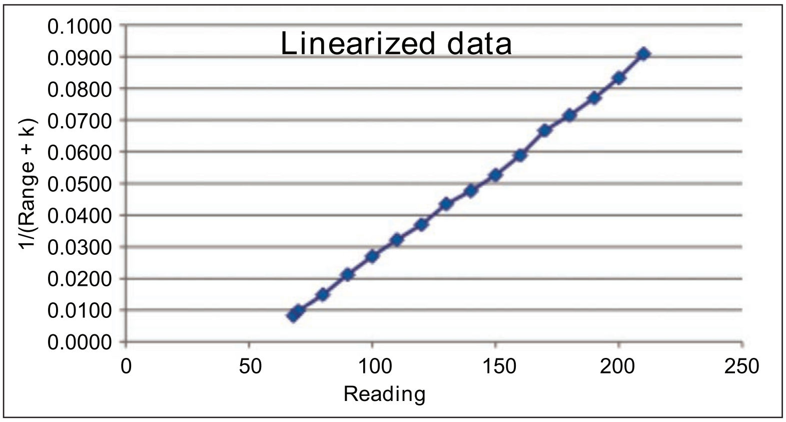


Figure 2. Linearized data after 'k'applied.

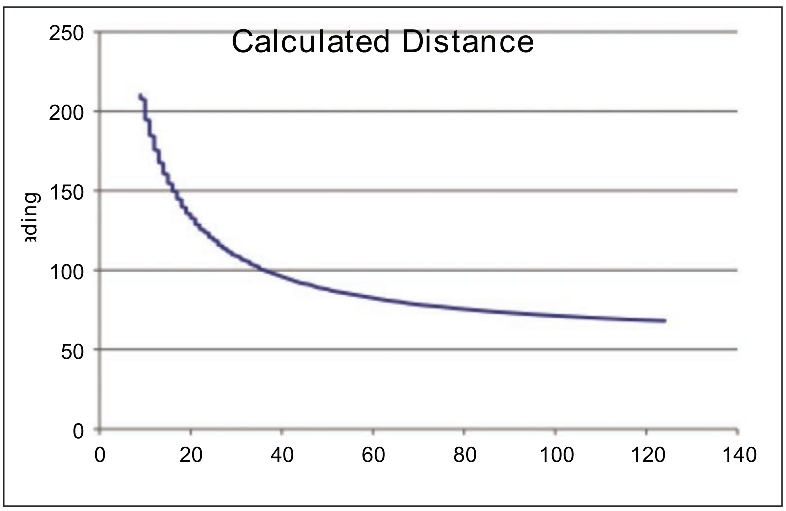


Figure 3. Calculated distance using voltage measurement and linearized formula.

coherent while I'm writing.

In short, no, you can't hack these to make them faster. In order to make the sensors reliable, there is a certain hysteresis required to stabilize the signal and to filter out some of the ambient light noise. These sensors can also be fooled by external light shining on them and also by what Sharp calls "glittery" surfaces like mirrors and glass.

Ordinary IR sensors will have all of the problems of the Sharp sensors above, and more. They are not modulated and won't reject ambient light noise. You don't want to go here; this is where line followers work well, but those can cover their sensors and surface to reject ambient light.

Using modulated IR beams and IR demodulators works well to get *yes/no* answers, but it is difficult to get range information from them reliably. Like #1 above, they will reject ambient light noise but they work by locking on to the modulated IR signal, and only drop their output when they have gotten a consistent signal over a period of time. You can adjust the detection range to a great degree by changing the intensity of the modulated signal, but it isn't range information.

In the size you want, creating another triangulation range finding method will be a challenge. Using a small CCD (like an optical mouse input) and a laser diode could achieve this, but it will be a non-trivial and very interesting project! You would have to tune it to a restrictive range window which could include very close ranges. I believe that you could use an optical filter to keep out ambient light and only trigger on the very bright laser reflection. An intriguing idea ... hmm. Will anyone take up this challenge? This sounds *very* interesting!

Any single transducer sonar sensor will not deal well with the range limitations you have in this competition, I believe. The Maxbotix sonars have a 15 cm minimum range before they will give any more range information. Dual element sonar's like the Parallax Ping))) get closer (10 cm) because they don't have to wait as long for the transducer to stop "ringing" from the transmitted pulse. The more expensive Devantech SRF04 (and others) have everyone beat with a range distance down to 3 cm, but still, they recommend that you wait 50 ms between pulses for all the echoes to settle down. And there you have the real limitation: the sonar range finders can trigger faster than the Sharp range finders, but all those bouncing sound waves can mess up readings if you push the times. Another problem is that sonar is not all that *tight* in beam width and corners *really* show up strongly to a sonar signal.

You certainly can roll your own sonar. I recommend that you do that just to see how it works and to get an appreciation for the technology and its limitations. However, you won't get anything better than what has been developed over the last 10 years by the clever folks at Devantech, Maxbotix, and others.

You lost me here. I don't know what an "overhead IR sensor" is.

Hmm ... books on the Micromouse competition. There is a nice site on Micromouse design and competition ideas at **www.micromouseonline.com**. I found no other books dedicated to this competition; a search on

www.amazon.com will find books that include that competition in their descriptions of robot competitions, though.

It is not essential to use only analog data input for your PID algorithms. However, PID *is* analog, so you will have to use many digital (yes/no) type sensors to simulate an analog spread of input data.

My advice to you with respect to designing a Micromouse competition robot is to look at YouTube videos and hunt those robots down on the 'net. Find out how their designers solved their problems. There are many commonly used algorithms that have been tried and proposed — some of them are truly fantastic! There is nothing in the competition rules that says that this is an easy task! Robotics is an inherently difficult subject to deal with, and there are very few masters of this genre of engineering.

Best of luck!

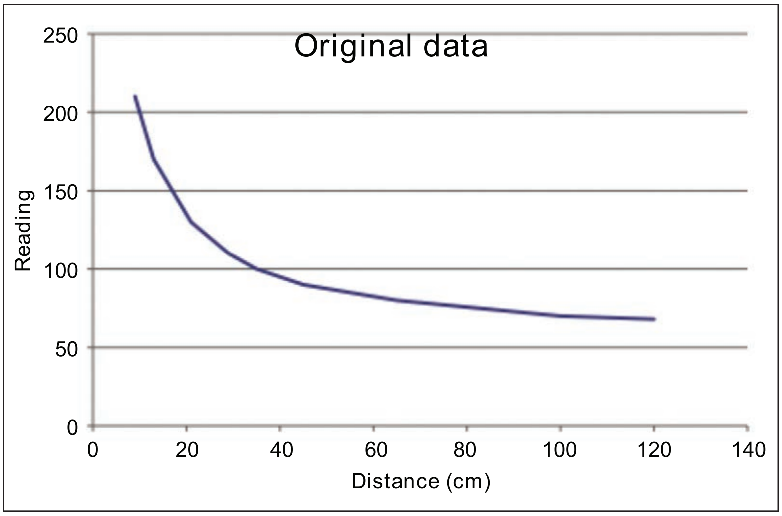


Figure 4. The originally measured data points.

Q. Is there a formula to plug in the value from a Sharp IR range finder and get the actual range out? I've been estimating based upon the values, but that just doesn't seem "scientific" enough.

— Dave

A. Yes, there is; **www.acroname.com** has a great article about just this type of calculation at **www.acroname.com/robotics/info/articles/irlinear/irlinear.html**.

Sharp starts by telling us that V = 1/(R + k). Doing some algebra to work our range (R) as the result of the voltage from the y = mx +b form (you do remember your college algebra, right?), we get the form R = (m'/(V + b')) – k where m' = 1/m and b' = b/m. Don't worry, you can get Excel to do this for you.

A friend of mine, read this and came up with an Excel spreadsheet that handles all of the calculations. You just plug in the numbers that you read from your IR range finder at carefully measured ranges and fiddle with the k constant until your line is straight. The secret is in the division of the voltage values and finding a k constant that will give a straight line. In **Figure 1** and **Figure 2**, I show the Excel curve of the measured voltage related to the actual range to the object and the final *linearized* (pardon me for inventing a verb) curve after I found a good k constant.

This is all nice, but what do I get out? **Figure 3** and **Figure 4** show the calculated ranges vs. the actual ranges read from the device. Those curves look very similar! It turned out that for *this* sensor, k = 2. This constant will be different for every sensor model, and will have some small variance between individuals of a particular model, but not much.

The key to success here is to take very careful measurements and spend some time working to find a good k constant. This approach works well if you round your constants to whole numbers. It works even better if you have the ability to use floating point math. Now that I've tantalized you with a spreadsheet so that you don't have to work all this math out by hand, I suppose that I should give it to you. You can download this spreadsheet from **www.servomagazine.com** under Mr. Roboto as linearizesharpcalc.zip. Enjoy!

Well, another month is gone and hopefully everyone has built something, or — failing that — has learned something after they put the fire out. Keep on sending me those cards and letters at roboto@servomagazine.com and I'll keep trying to answer your questions! Until later ... **SV**

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**.

— R. Steven Rainwater

FEBRUARY

5-9 **APEC Micromouse Contest**
Orlando, FL
Speedy autonomous mice robots solve mazes.
www.apec-conf.org

17-19 **Motorama Robot Conflict**
Harrisburg, PA
Remote control vehicles destroy each other.
www.nerc.us

23-26 **Pragyan**
National Institute of Technology, Trichy, India
Events include Traffic Rush, Fix the Android, Diner Dash, and Lanka Reloaded.
www.pragyan.org

MARCH

10 **Trenton Computer Festival Robotics Contest**
Ewing Township, NJ
Every year is different, but robot events usually include maze navigation, precipice avoidance, and Micromouse.
www.tcf-nj.org

10-11 **RobotChallenge**
Vienna, Austria
Events include Parallel Slalom, Slalom Enhanced, Sumo, Mini Sumo, and Micro Sumo.
www.robotchallenge.org

16-20 **Apogee iStrike**
BITS Pilani KK Birla Goa Campus, Zuarinagar Goa, India
Autonomous ground robots must follow traffic signs while avoiding obstacles.
www.bits-apogee.org

17-18 **METU Robotics Days**
METU Culture & Convention Center, Turkey
Events include Sumo, Mini Sumo, Slalom, Triathlon, Garbage Collection, Mine-Sweeper, and others.
http://topluluk.odturobotgunleri.org.tr

20-22 **DTU RoboCup**
Technical University of Denmark Copenhagen, Denmark
Events include line following and wall following.
www.robocup.dtu.dk

24 **CIRC Central Illinois Bot Brawl**
Lakeview Museum, Peoria, IL
Lots of events including line following, line maze, Sumo, RC combat, and Best of Show.
http://circ.mtco.com

24 **Greater Philadelphia Sea Perch Challenge**
Drexel University, Philadelphia, PA
Tethered underwater ROV event.
www.phillyseaperch.org

24 **Harrisburg University Pennbots**
Harrisburg University, Harrisburg, PA
Maze solving plus remote control vehicle combat.
http://web.me.com/wjbechtel/Robot_Competition/Welcome.html

30 **Trinity College Fire Fighting Home Robot Contest**
Trinity College, Hartford, CT
Contest runs through April 1. Autonomous robots must navigate a mock house, locate a fire, and extinguish the flames.
www.trincoll.edu/events/robot

NEW PRODUCTS

Waterproof Servos

Joining the HS-646WP and HS-5646WP, the HS-5086WP completes the trilogy in Hitec's class of waterproof servos. With high voltage capability, heavy-duty metal gears, and a ball bearing supported output shaft, the HS-5086WP is among the most durable and reliable micro servos for wet conditions. Possessing the same size as Hitec's popular HS-85MG and the industry's first IP67*rating, this mini servo will keep an RC vehicle, boat, aircraft, or robot watertight, regardless of weather conditions or the surroundings.

Features include:

- Metal Gear with MP First Gear.
- Top Ball Bearings.
- Three-Pole Cored Ferrite Motor.
- Two-Cell LiPo Capability.
- Programmable Circuit.

The estimated retail price is $49.99.

For further information, please contact:

Hitec — Website: **www.hitecrcd.com**

Super Duty Belt Drive Pan System

ServoCity introduces their all new Super Duty Belt Drive Pan System. This system offers a simple way to reduce the speed and increase the torque of a gear motor. The pan offers a 6:1 reduction through the XL series timing pulleys and timing belt. The all-aluminum framework is able to handle up to 40 lbs. Dual ABEC-5 ball bearings support the 1" OD hollow stainless steel shaft which allows wires and cables to be routed through the axis of rotation. The toothed Kevlar belt ensures minimal backlash and isolates noise transmitted from the gear motor. There are 1/4-20 tapped holes in the 1" bore clamping hub to allow for easy mounting of various attachments. The Super Duty Belt Drive Pan system is compatible with 3-12V standard gear motors or 6-12 VDC precision gear motors. The unit comes mostly assembled, with only the pulley and belt requiring assembly once the gear motor is mounted. The gear motor is sold separately.

Vertical Shaft Worm-Drive Gearbox

ServoCity also introduces their new Vertical Shaft Worm-Drive Gearbox. This gearbox utilizes a 30:1 ratio worm-drive reduction for applications that require extremely slow and smooth rotational motion. The worm-drive design minimizes backlash and eliminates back-driving the gearmotor so a position can be held even when power is not applied.

The precision ground 3/8" stainless steel output shaft is supported by dual 3/8" ABEC 5 ball bearings to support a load from any orientation. The 1/4" ABS plastic and aluminum structure provides a rigid framework without adding unnecessary weight and is easily mounted to any flat surface by utilizing the base mounting tabs. The Vertical Shaft Worm-Drive Gearbox is ideal for turn-tables, time-lapse systems, and low speed applications that require high precision and torque.

For further information, please contact:

ServoCity — Website: **www.servocity.com**

7-in-1 Rechargeable Solar Transformers

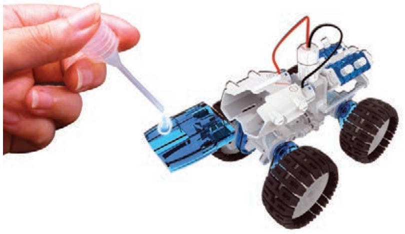
OWI, Inc., has added the 7-in-1 Rechargeable Solar Transformers to their line of construction kits which are available for all levels of experience.

This solar science kit encourages builders to exercise their creativity and imagination. The 7-in-1 kit comes with all snap plastic parts (no screws, tools required), solar panel, and accessories. It is easy to

assemble the seven different modules: two power charging stations and five different vehicles.

The kit comes with a full time fueling station. The transforming rechargeable station has two output plugs: one for solar recharging and the other for battery recharging. The battery charging station can bring the selected unit to full power in about five seconds, and the solar recharging station will fully recharge in approximately 120 seconds. Run time is approximately 50-80 seconds. The 7-in-1 Solar Rechargeable Solar Transformers allows the user to experience energy that can be recharged and stored in a power source, and learn its principles and knowledge through assembly. The suggested selling price is $26.95.

Salt Water Fuel Cell Monster Truck Kit



OWI, Inc., has also added a new Salt Water Fuel Cell Monster Truck kit. Building on the success of OWI's Salt Water Fuel Cell Car, this second generation is bigger and badder. The Salt Water Fuel Cell Monster Truck kit is equipped with four-wheel drive mechanical construction. It easily handles different types of terrain by its twisted car body design. It has two transformable engine blocks: V engine or flat engine. The transparent case design allows children to see the rhythm of the piston in a vertical or horizontal position. Each monster wheel can change its angle individually, and the truck can change its height and shape. Forward and reverse are no problem; simply switch the "positive and negative" connectors. The kit is fueled by salt water. This Monster Truck gives children a chance to learn about new forms of clean energy while building and powering their own kit.

After activating the fuel cell module with a salt water mixture, the magnesium metal sheet (three sheets included) can operate the car for about five to seven hours continuously. If you want to park the vehicle, simply take out the fuel cell module, and rinse with tap water and dry.

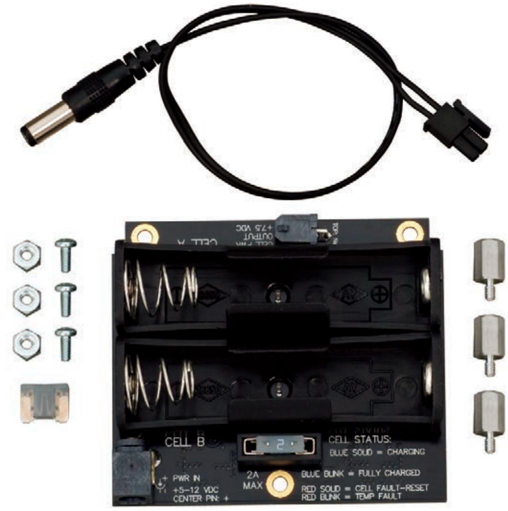
All the materials used in this kit are environmentally safe and clean. Non-toxic substances are used so there are no disposal concerns. It will not produce heat, so it is safe for child's play. Dimensions of the assembled product are 3.52" (H) x 3.9"(W) x 4.72"(L). The suggested selling price is $25.95. For further information, please contact:

OWI, Inc.

Website: **www.owirobot.com**

Li-ion Boe-Bot Power Pack Charger

The new Li-ion charger from Parallax makes it possible to no longer have to remove rechargeable AA cells and



place them into a separate charging station, or to have to replace non-rechargeable alkaline cells. Simply plug in a 7.5 VDC 1A wall transformer, and the charging circuitry does the rest automatically. This pack/charging solution can replace up to several thousand AA alkaline batteries over the Li-ion cells' lifetime.

Two Li-ion 18650-size cells and a power supply are required and are not included.

Features include:

- Holds two rechargeable/replaceable 3.7 volt Li-ion 18650 cells.
- Provides up to six hours of continuous motorized operation (depending on cell capacity).
- Rechargeable Li-ion cells don't need to be removed from your Boe-Bot for recharging — charging circuitry is built into the board.
- Multiple LED indicators provide charge readiness information for each individual cell; the status key for the LED indicators is printed on the board.
- Aggressive holders retain cells in any board orientation and in moderate shock environments such as mobile robotic applications.
- Dedicated circuitry continuously monitors the charging process to ensure safety, efficiency, and to maximize the number of charge/discharge cycles of each cell.

Price is $44.99.

Dual Motor Driver MC33926

Also available from Parallax is the dual MC33926 motor driver carrier which is a breakout board featuring two Freescale MC33926 H-bridge ICs. It can supply up to almost 3A continuous current per channel to two brushed DC motors at 5–28V, and it can tolerate peak currents up to 5A per channel for a few seconds. This makes it a useful general-purpose motor driver for medium-sized DC motors and for differential-drive robots that use such motors. The MC33926 supports ultrasonic (up to 20 kHz) pulse width modulation (PWM) of the motor output voltage which eliminates the audible

switching sounds caused by PWM speed control. A current feedback circuit for each motor outputs an analog voltage on its respective FB pin that is proportional to the output current.

Features include:

- Breakout board for Freescale's MC33926 full H-bridge.
- Delivers 3A continuously to each of its two motor channels.
- Current feedback, and under-voltage protection.
- Over-current and over-temperature protection.
- Reverse voltage protection on motor voltage (logic voltage does not have reverse protection).

Price is $29.99.

Motor 12V Geared Motor

Also new to Parallax is a powerful 12 VDC geared motor with 200 RPM output that can be the muscle in your next robot. Manufactured by Hsiang Neng, these motors are solid and a direct replacement for the 7.2V motors used in other Parallax compatible kits.

Features include:

- 200 RPM Output Shaft (±10%).
- Output Shaft Diameter of 6 mm.
- 60 mA Average Current @ No Load.
- 4.5V-12V Range (6V nominal).
- Spur Gear Design.

Price is $19.99.

For further information, please contact:

Parallax, Inc. Website: **www.parallax.com**

Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to:

newproducts@servomagazine.com

Robotics Showcase

bots IN BRIEF

NEW NAO

Before you've even managed to save up for one of the original Naos, Aldebaran Robotics has come out with an entirely new, even more awesome version.

So, what exactly do you have to look forward to?

- Nao is skinnier. Longer, thinner arms give Nao better reach and more working space in which to grasp things.
- There's now a full-fledged Atom processor inside Nao. (Helloooo multitasking!)
- Speaking of multitasking, two HD cameras provide parallel video streams, helping Nao get better at face and object recognition — especially in bad lighting.
- "Nuance" voice recognition helps Nao pick key command words out of sentences.

But wait! There's more! "On top of this new hardware version, we shall be delivering new software functionalities like smart torque control, a system to prevent limb/body collisions, an improved walking algorithm, and more. We have capitalized upon our experience and customer feedback in order to deliver the most suitable and efficient platform. In terms of applications especially at high school level, we are focused on educational content while, when it comes to improvements in personal well-being, we are working on developing specialized applications," explains Bruno Maisonnier, founder of Aldebaran Robotics.

Want one? Of course you do! It looks like you'll still have to go through the developer program to get one, though, and that'll run you somewhere around $5k.

KICKASS K-MAX

Helicopters are typically the most reliable way to get supplies to some of the more remote outposts in Afghanistan, for example, but flying resupply missions is dangerous work. Now, some of those aerial resupply jobs are being taken over by an unmanned K-MAX helicopter.

The K-MAX is an unmanned (or optionally manned) conversion of the Kaman K-MAX aerial truck, modified for autonomous operations by Lockheed Martin. The K-MAX has that "aerial truck" moniker because it was designed from the ground up for cargo lifting with intermeshing rotors that allow it to lift three and a half tons of cargo (more than the weight of the helicopter itself) up to 250 miles.

In a test last year, the K-MAX went from "boy, this would be great if we could get it to work" to "now in preparation for sustained operations" after it accomplished an autonomous cargo delivery to an unspecified location in southern Afghanistan. Using the K-MAX instead of a manned helicopter protects human crews, of course, but also allows for more missions to be flown more frequently, because robots don't get tired and are generally pretty good at flying in the dark.

K-MAX has recently been worked over by the Marine Corps, and if it checks out, the Army, Navy, and Air Force might all start to invest in an entire fleet of little robotic delivery copters.

bots IN BRIEF

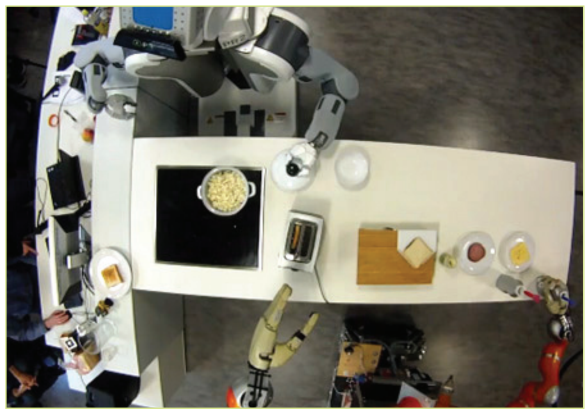
MAD MorpHex

Yep. There's a hexapod tucked away inside this transforming sphere. Designed by Zenta, this little guy could do pretty well at a hexapod dance-off competition or it could be armed and made operational with giant death lasers. It's not clear if it's ever going to be turned into a kit, but keep in mind that it uses a wallet-busting 25 separate servos, so you'd better start saving now. And win the lottery. A few times. And, as if transforming from a hexapod into a sphere and back again wasn't enough, there's a plan to teach MorpHex how to roll itself along the ground.

SKILLED LABOR

James and Rosie — a lovely robotic couple at the Technical University of Munich — are well known for their delicious pancake and sausage breakfast rituals. Now, their skills have expanded to include both sandwiches and popcorn.

Recall that none of these steps are pre-programmed. The robots are able to understand what steps go into something like making popcorn, and break those steps down into actions. Kinda like TUM's PR2 wasn't explicitly instructed to go turn the stove on and off; it just knew that popcorn required using the stove, and that the stove needed to be turned on. So, it did all the localization and navigation and manipulation of the stove controls by itself.

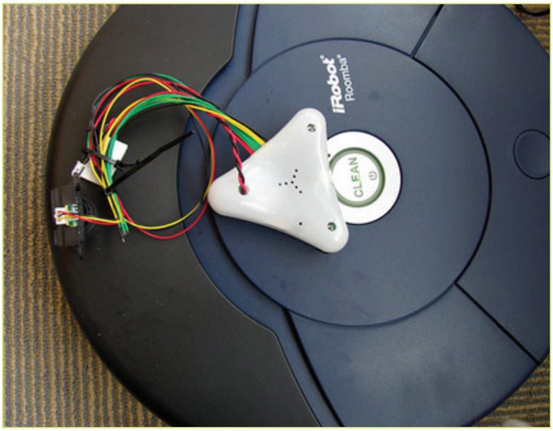


The reason that this research is so important is that we don't want to have to be endlessly providing robots with instructions for every last thing they're supposed to be doing. Giving robots the ability to take a complex task and autonomously infer all the intermediate tasks that it can, then execute one at a time means that you'll be able to say, "Make me a sandwich" or "Do my laundry" or "Clean the house" or "You know what, go get everything done while I take a nap." This way, the robot will just go and do it, no questions asked.

DRAGON IN THE CLOUD

Kombusto — despite his extremely dragony appearance — doesn't have to exist in a beclawed corporeal body. Since he lives in the Cloud — even without a fancy $1,000 robot — kids are still able to have a dragony friend on their (Android) smartphone. The fancy name for this is "blended reality," and it's a powerful tool for education since it removes the traditional hardware constraints that come with robots.

The other advantage of Cloud robotics is that every time Kombusto gets smarter or learns something new, all of the other incarnations of Kombusto in the hands of other kids can directly benefit from the upgrades. While $1,000 may seem expensive to you (and it is expensive), for an institutional research robot this is dirt cheap. But, does he like to cuddle?



BRAINLINK POWER

Roomba is a reasonably clever robot. Heck, for a vacuum, it's downright brilliant. Compared to other robots, though, it's lacking some sensors and skills that are starting to become basic. Roomba could use a new brain, and since it's a robot you can actually just go and give it one.

Brainlink is a little piece of hardware that can augment (or replace) the brain that's currently powering your robot. Any robot actually, as long as it's controllable with an IR remote or a serial connector or some other common type of interface. Brainlink itself (the plastic triangle thingy in the photo) talks to your Android phone (or a computer) via Bluetooth, enabling programming and wireless control of whatever it's attached to.

In addition to providing a new programming interface for robots that may not come with one, Brainlink can also be configured to use a wide variety of sensors. A three-axis accelerometer and a light sensor are built in, and there's a whole heap of digital and analog connectors that make it easy to plug in, for example, proximity sensors to keep your Roomba from running into stuff.

The overall idea with Brainlink is that there are a bunch of robots out there available for very cheap with fundamentally sort of decent hardware, but no easy way to get them to do what you want. Brainlink provides these less-than-clever robots with a new level of usefulness that makes them suitable for anyone with desire and some basic programming skills to mess with. A Brainlink module will set you back $125, but I'd say that's not too much to ask for a brand new brain, right?

FEMALE INTERACTION

Meet an abstract female humanoid robot named OriHime. It was created by Kentaro Yoshifuji — a four-year Engineering student at Waseda University — who felt compelled to build a new kind of communication robot. After graduating high school, Kentaro took an interest in artificial intelligence. Therapeutic robots like the baby seal Paro were being introduced in hospitals, so he thought maybe a robot could become a kind of artificial friend. After volunteering in hospitals and interacting with people, he realized that they really wanted to connect with other people.

While robotic wheelchairs, beds, and other equipment give more freedom to hospital patients, many still feel isolated and lonely. Young people with chronic illnesses can spend a lot of time in hospitals, and because they don't attend school they don't get the chance to make many friends. As the elderly population increases and the traditional family home becomes fragmented, people tend to communicate less and less. His solution was a robot equipped with a camera so that a human operator could see a live video feed using an Internet connection. By building a humanoid robot "avatar" capable of a variety of expressions and gestures, the feeling of communication could go beyond a simple telephone call to a sense of physical presence.

Design-wise he says he had no knowledge of humanoid robots, and didn't take much interest in science fiction stories growing up. Everything except the servo motors had to be built from scratch through trial and error. Its face was kept simple on purpose, with eyes based on a feline. Rather than showing emotion through the face — which he admits looks a bit scary — the robot's whole body conveys the operator's mood. As a result, even a blank face begins to take on a certain character. He studied dancers and mimes at a festival and tried to incorporate the feeling of their movements into the robot. Technical details are scant, but it was actually completed in 2009, stands approximately 60 cm (2') tall, and has 24-26 degrees of freedom, depending on the configuration.

I SPY

Ever wonder if you've been spied on by a surveillance drone? Maybe it looked like a hummingbird. Or an insect. Or, maybe it was just really high up. Maybe there's one looking in your window right now, and if so, there's no law that says it shouldn't.

In a recent article in the *Stanford Law Review*, Ryan Calo discusses how domestic surveillance drones would fit into the current legal definitions of privacy (and violations thereof), and how these issues could influence the future of privacy policy. In a nutshell, surveillance robots have the potential to fundamentally degrade privacy to such an extent that they could serve as a catalyst for reform.

Domestic surveillance robots aren't as much of an issue now as they could be, thanks mostly to the stick-in-the-muddedness of the FAA that keeps unmanned aircraft from doing anything exciting. Eventually, though, that's going to change, and there are already precedents (legal ones) for how domestic agencies might (read: will) start using robots. Basically, there seems to be essentially no legal restrictions which would prevent the police from having drones flying around all the time, watching people.

Clearly, this is something that we as a society should discuss, and we may decide this kind of surveillance should be illegal or at least restricted to some extent — especially since it's getting easier and easier to build or buy camera-capable flying robots. In the near future, celebrities will be constantly surrounded by a swarm of face-reading, photo-snapping autonomous robots that will necessitate the development of anti-surveillance drone drones, loaded up with little miniature air-to-air missiles (which themselves are little flying robots).

Of course, all this goes beyond surveillance and drones. We've got these same sorts of legal issues popping up all over the place with regard to robotics, as technology fast outpaces the limited amount of foresight that was employed when coming up with policies meant to manage current technological issues as opposed to future ones. There's a risk that reactionary (as opposed to proactive) policies could seriously undermine the robotics industry which is why forethought is so important. Be sure to look up and read the rest of Ryan Calo's article, "The Drone as Privacy Catalyst."

TROOP TESTER

A group of experts who come from Sydney University have created a robot that is being used to test troops who may run into terrorists. Sent to a Marine base in Virginia's Quantico, the Marathon Targets are trained to "think" and because they are autonomous, flee as fast as a human if one of their group is shot. They even have enough smarts to seek shelter.

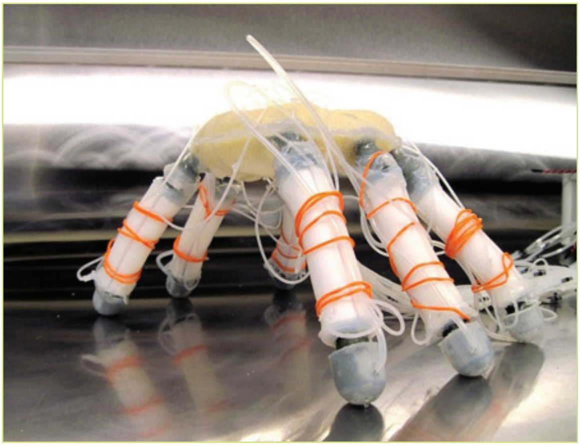
The company received a $57 million contract with the USMC after showing its lifelike armored-plated T2 prototype. They also have another Terrorbot with four wheels for use in rough terrain.

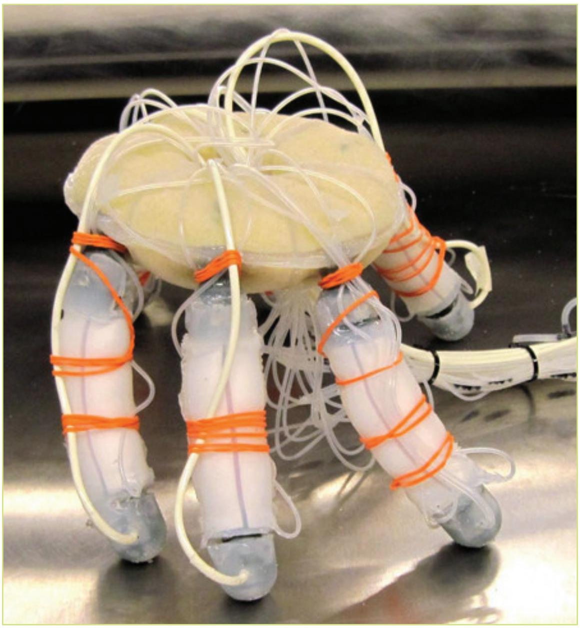
(NOT) ANGRY BIRDS

RobotGrrl has new totally DIY-able interactive robotic birds called RoboBrrds. If you've always wanted to throw yourself bodily into the world of Arduino-powered DIY robotics, this is a great way to go.

You can build your very own RoboBrrd. It'll take about a week's worth of on-and-off work and you'll likely need to order some electronics, and if you don't know how to solder, well, here's a great excuse to learn!

Once you get your very own RoboBrrd up and running (or even if you don't), you can share it with the world (or at least with fellow robotics geeks) every Thursday night at 8 pm EST through a Google+ video hangout. You should also remember that RobotGrrl is doing all of this out of the goodness of her (robotic?) heart, and there's a handy little "donate" button on her website (**robotgrrl.com**) should you wish to help inspire future generations of roboticists.





GETTING A GRIP ON WALKING

This is the Jamming Modulated Unimorph Hexapod (a.k.a., JHEX) — a prototype soft robot that's based on the coffee grounds and balloon system developed by iRobot, Cornell University, and the University of Chicago. The basic idea behind the whole jamming thing is that by changing the pressure inside a flexible container filled with a granulated material, you can make the particles of the material either flow around each other or "jam" together, causing the container to transition between softness and rigidity. JHEX can walk by switching its legs between these soft and rigid states, or it can go from completely rigid to completely soft. We've been seeing all of these soft robots recently thanks in large part to DARPA and their ChemBots program.

The goal of the Chemical Robots (ChemBots) program is to create a new class of soft, flexible, meso-scale mobile objects that can identify and maneuver through openings smaller than their dimensions and perform tasks once entry is gained. The program seeks to develop a ChemBot that can perform several operations in sequence. It should travel a specified distance and traverse an arbitrarily shaped opening much smaller than the largest characteristic of the robot itself. Once through the opening, it will reconstitute its size, shape, and functionality, and travel again to perform a task using an embedded payload.

Just about all of the Chembot prototypes that we've seen so far have been tethered to a power source (like compressed air) which means that you've got a way to disable these things if they start oozing (or jamming) towards you. Getting power sources and payloads onboard the robots themselves isn't going to be easy, but based on the creativity and progress that we've seen so far, it looks like DARPA will get what it wants, and we'll have something new to feel vaguely uncomfortable about.

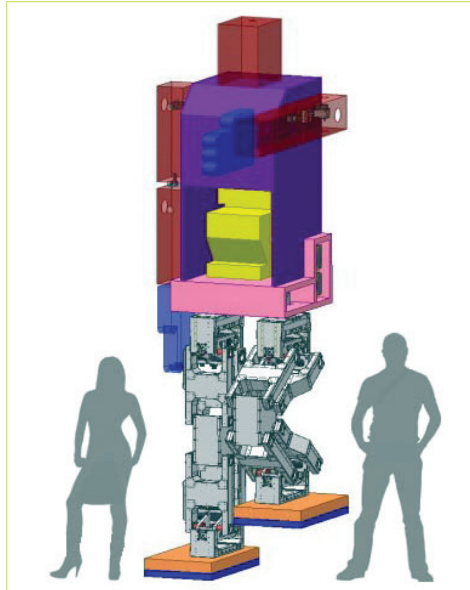
AVA MAKING ROUNDS

One telepresent/iPad bot has been built for Mt. Sinai Hospital in Toronto to assist Dr. Mirek Otremba and the rest of the staff. iRobot has already teamed up with InTouch Health — a robotics company that specializes in medical conferencing by remote — so we expect that there will soon be an AVA appearing at a health care facility near you. While winsome-looking, beeping, and scurrying robots that interface with your computer are already on the market, iRobot's AVA will put an iPad face and brain on the droid's navigational skills, microphones, speakers, lasers, and senors. Acting as a doctor's consultant or assistant is a key early focus for AVA.

ROBO THERAPY WORKING?

A team from the Santa Lucia Foundation in Rome, headed by Dr. Giovanni Morone, recently completed a trial study to see if those who have a stroke can benefit from robot-assisted therapy. The study took two years and consisted of 48 nonambulant patients who became involved after 20 days from the time of the strokes. They found that using the robotic assistance helped those that had high motor impairment. If you would like to study their findings, it was published in *Stroke: Journal of the American Heart Association* and can be found at **http://stroke.ahajournals.org/content/early/2011/12/15/STROKEAHA.111.638148.abstract**.

TALL ORDER



Apparently, Hajime Sakamoto — the president of the Hajime Research Institute (a Japanese company that has been building humanoid robots since 2002) — is unhappy with the world famous 1:1 scale Gundam statue. It's nice and all, but it's still just a statue, and he wants to build the real deal. In 2009, he built a 210 cm (7 ft) tall robot – one of the tallest in the world – and now he's attempting to build a working four meter (13 ft) tall version. The robot will even sport a built-in cockpit. Next, he plans to build one that is eight meters (26 ft) and if all goes according to plan, he'll eventually build one that is 18 meters (59 ft) tall – the size of a Gundam mobile suit. It may sound impossible, but a dream is a dream.

The company is looking for sponsors to help them complete the current project, and is working with NKK Kyousei and contractors to build the parts. In the meantime, you can become a fan of the project on its official Facebook page.

EOS E-ONE

E-One is a remote monitoring robot developed by EOS Innovation — a French company founded in March 2010. The company says the robot is still only experimental, but is intended to patrol on its own and alert its owner if it detects anything unusual. The robot may also serve the elderly and disabled in the future, and could function as a telepresence robot. Although it looks somewhat similar to NEC's PaPeRo, its face displays video.

At 60 cm (2') tall, the E-One is relatively small compared to similar robots, but has a low weight (8-10 kg [22 lbs]) and a decent battery life of four hours. It comes equipped with two high-resolution cameras (one of which has a fish-eye lens), as well as a mobile (pico) projector. It has two speakers and two omnidirectional microphones, and can communicate over Wi-Fi and USB. For obstacle detection, the robot uses eight ultrasonic sensors and four infrared sensors.

"Right now, the biggest weakness in robotics is battery life. The batteries have to last for several hours, so our robots must use energy efficiently. Speech and visual recognition are also two domains where there's a lot of work to be done. In my opinion, we're heading towards specialized robots that can communicate with one another to make things easier. Whether it's a robot vacuum cleaner, lawn mower, or butler, they'll be able to take orders and divide up the work. But a machine that can replace a person, that's not for right now. I think we should make simple robots that are optimized for specific tasks." — David Lemaitre, Director and Founder, EOS Innovation.

The company has also developed a smaller, semi-autonomous surveillance robot called E-Vigilant. The idea is to have the robot patrol on its own, but if it detects something suspicious a human operator would take over. Due to its low profile, the robot could follow an intruder undetected. The E-Vigilant is due to be commercialized in 2012.

Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.

SMARTPHONES AND SPHERES

International Space Station researcher Mike Fossum — commander of Expedition 29 — puts one of the Smart SPHERES through its paces. The inset image shows the Samsung Nexus S™ handset that helps turn the SPHERES into mobile data acquisition assistants for space explorers. *Photo courtesy of NASA.*

In November last year, a free-flying robot on the International Space Station successfully gathered and delivered motion data to its astronaut handler for the first time via a new smartphone controller.

The Human Exploration Telerobotics project — one of NASA's new, high-value Technology Demonstration Missions — equipped the compact, free-flying satellites known as Synchronized Position Hold, Engage, Reorient Experimental Satellites, or SPHERES with a Samsung Nexus S handset that features Google's open-source Android platform.

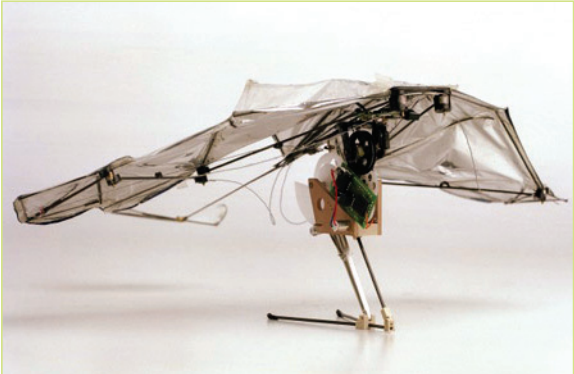
Each volleyball-sized SPHERES has its own onboard power, propulsion, computing, and navigational software. Adding the smartphone transforms the satellite into a free-flying robot, or "smart SPHERES" complete with a compact, low-power, low-cost embedded computer and built-in cameras and sensors to enhance and expand robotic operations.

Minor modifications were made to the smartphones, including removing the GSM cellular communications chip to avoid interference with station electronics, and replacing the standard Lithium-Ion battery with AA alkaline batteries. Otherwise, the smartphone is identical to the off-the-shelf consumer device.

The ongoing experiment demonstrates how Smart SPHERES can serve as remotely operated assistants for astronauts in space. In coming months, these compact assistants will conduct interior station surveys and inspections, capturing mobile camera images and video. NASA also plans to simulate external free-flight excursions and in time will test whether the robots can handle other, more challenging tasks.

"The tests that we are conducting with Smart SPHERES will help NASA make better use of robots as assistants to and versatile support for human explorers — in Earth orbit or on long missions to other worlds and new destinations," said Terry Fong, project manager of the Human Exploration Telerobotics project and Director of the Intelligent Robotics Group at NASA's Ames Research Center in Moffett Field, CA.

The Nexus S phone is the first commercial smartphone certified by NASA for use on the space station. Each smartphone is connected to a SPHERES free-flyer via a cable. A wireless network connection (Wi-Fi) to the space station's computers provides the data path to the ground. NASA anticipates using other types of smartphones on the station in the future.



WINGING IT

Gliding is a very efficient way for getting from point A to point B. Jumping is a very efficient way of getting into the air at point A, especially if there are a bunch of obstacles between point A and point B that you need to be airborne to make it over. Grasshoppers have been doing this for like a hundred million years, and roboticists at EPFL are starting to design their robots with the same kind of jumping talents and expandable wings as our orthopteran friends.

The jumping part and the crawling around on the ground part is somewhat impaired by the bot's giant wings, which is why getting this whole folding thing figured out would be pretty cool.

Locusts aren't the only creatures with wings that cleverly fold up. EPFL is also trying out a system based on bats. There's also some super secret third bio-inspired design that might be used as a basis from which to create a gliding robot. Wonder what that will be ...

MOK-i SEE, WON-i DO

Mokwon University (South Korea) held a four-day Robot Festival in November '11. This is a yearly event that began in 2006 to showcase student robot projects. The 20 or so examples demonstrated last year included some that could be controlled using a smart phone, autonomous vehicles that used both stereoscopic vision and 3D vision systems, service robots, small bipeds, and guides for the visually impaired. The humanoids in the photo (Mok-i and Won-i) have attracted several prospective students to the University's booth at many of the recent college and university fairs. The lab boasts a 100% graduate employment rate for the past two years. Mokwon University is located in Daedeok Innopolis — a technological innovation hub in Daejeon made up of several universities and research institutes including KAIST.

WHAT A SITE!

Check out the new website called **MyRobots.com** which aims to be a sort of social network and Cloud communications system for consumer robots and other "smart" household objects.

COMBAT ZONE

Discuss this section in the SERVO Magazine forums at http://forum.servomagazine.com

Featured This Month:

www.servomagazine.com/index.php?/magazine/article/february2012_CombatZone

The History of Robot Combat: BattleBots

● by Morgan Berry

In the last issue, we discussed the birth of robot combat with Critter Crunch at MileHiCon, and the sport's growth into the popular Robot Wars competition and British television series. This month, we examine the transformation of the sport into a nationwide phenomenon.

For years, robot combat in the United States was a niche sport reserved for relatively small groups of builders. In the course of less than two years, however, the BattleBots™ competition grew from another middle sized competition to a nationally broadcast show on Comedy Central. Television audiences had never seen anything like it; BattleBots was a complete novelty — and Americans could not get enough. What led to the show's success? In an online Q&A session with fans of the show on the *Washington Post's*

website in 2000, cofounder Greg Munson chalked it up to pure human nature: "People like to see things get destroyed," he said in an answer to one question. History agrees; robot combat is really just a slightly more humane version of gladiator fights in ancient Rome. Whatever the reason for the appeal, it is certainly true that the BattleBots television show brought huge mainstream popularity to robot combat.

The Start of BattleBots

As I discussed in last month's issue, the late 1990s were a turbulent time for robot combat. Marc Thorpe, the founder of the popular Robot Wars competition, was entangled in legal battles over the Robot Wars brand. Several events in San Francisco, CA were planned and then cancelled due to legal issues. Eventually though, a pair of veteran builders — Trey Roski and Greg Munson — were given the legal go-ahead to host an event. A few months later in August 1999, the first official BattleBots competition was held in Long Beach, CA.

The event had three weight classes: Kilobots (25-55 lbs wheeled, 25-83 lbs legged), Megabots (56-109 lbs wheeled, 84-164 lbs legged), and Gigabots (110-200 lbs wheeled, 165-300 lbs legged). Notice that legged bots — like the impressive 450 lb Mechadon from Mark Setrakian (**Figure 1**) — were allowed to be much heavier than their traditional wheeled counterparts. Because of the difficulty involved in building them, only a few legged bots were entered in the competition.

There were the obligatory

FIGURE 1. Mechadon.
Photo courtesy of teamkiss.com.

FIGURE 2. Tentomushi.

Thanks for the Memories

Believe it or not, it has been over 10 years since BattleBots first aired on television. Some of the veterans of the competitions were kind enough to share their memories of participating in the events.

Terry Ewert, driver of superheavyweight champion Son of Wyachi, remembers the challenge and fun of the competitions: "The days were really long; [we did not get] much sleep during the event week. But we were operating under the motto of work hard and play even harder."

Michael "Fuzzy" Mauldin, driver of FrostBite (along with many other ice-themed bots), gave us a vivid account of his proudest moment: winning against Brian Nave's Phrizbee Ultimate during Season Five. "Phrizbee Ultimate had been tearing its way through the brackets, demolishing robot after robot (and wreaking havoc on the arena walls, too). After the first big hit, IceBerg's plow blade tore off and flew across the arena. But the huge hit disconnected Phrizbee Ultimate's weapon batteries, and from then on, our weaponless superheavy pushed Phrizbee around the arena. Then, Iceberg pushed Phrizbee up against the screw hazards and Phrizbee toppled upside-down. It wasn't our first knock-out, but it was our most satisfying."

Jason Bardis, who drove Dr. Inferno Jr. (which took home two lightweight championships), recalled the spirit of camaraderie that surrounded the competitions: "My best memories ... are the many instances where a competitor helped another fix their robot, just before they were to fight each other, so that it would be a fun, fair, and exciting fight. My largest and closest set of friends comes from the robot combat community. Whenever I go on a trip, I meet up with my bot friends."

FIGURE 3. Chin-Killa.
Photo courtesy of robotcombat.com.

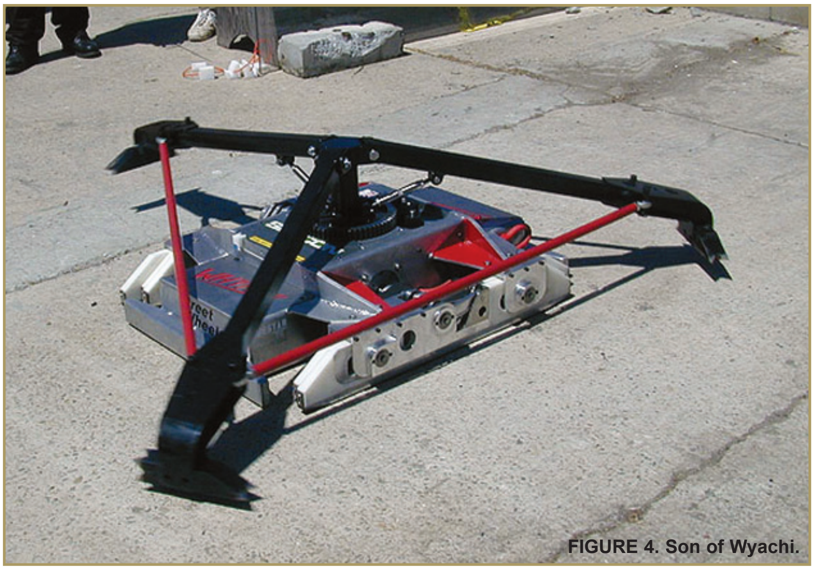


FIGURE 4. Son of Wyachi.

funny/silly robots, like Stuffie — a megabot that sported the shell of a remote controlled jeep with a teddy bear at the wheel — and Tentomushi (**Figure 2**) — an adorable ladybug themed bot that won over the crowd to claim victory in two separate matches.

Popular robots from the BattleBots television show, such as Carlo Bertocchini's BioHazard, or Deadblow — Grant Imahara's bot that has made several appearances on Discovery Channel's MythBusters over the years — also took part in this original competition.

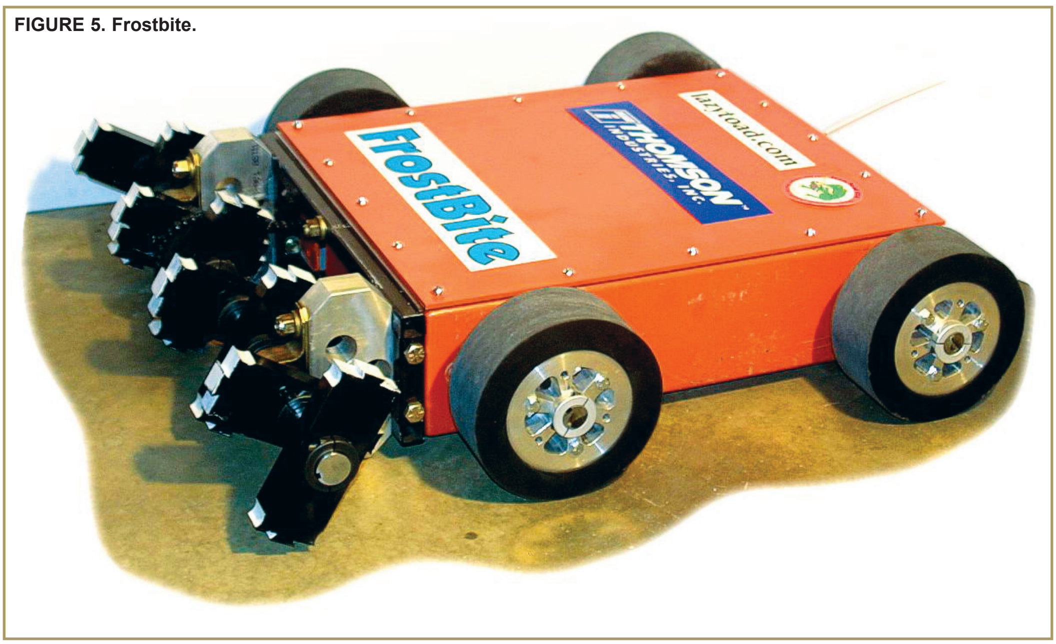
A few months later in November 1999, a second competition was held. This event used the weight class names of Heavyweight and Superheavyweight that fans of the television show were familiar with. A pay-per-view broadcast of this event gave television audiences a taste of the young competition — and soon America was hungry for more.

The Television Series Rockets to Success

The pay-per-view special generated a lot of interest in BattleBots and robot combat as a whole. Soon, it was announced that Comedy Central intended on adapting the competition into a regular television series. The show premiered in May 2000 and was soon a hit with television audiences. The premiere episode boasted some of Comedy Central's highest ever ratings at the time. The season was split into 13 episodes but in reality, the tournament took place in a single weekend in San Francisco.

"Spectacle" was a uniting theme in the program. Playboy Bunnies and former Baywatch actresses such as Carmen Electra were featured as correspondents, as well as comedians Bill Dwyer and Randy and Jason Sklar. The show also boasted popular scientist Bill Nye as a "technical expert." These

FIGURE 5. Frostbite.



famous faces added an air of relatability to the general public, but the competitors were mainly veteran builders.

To add more excitement to the matches, arena hazards operated by “Pulverizer Pete” like the pneumatic rams known as “Hell Raisers” could spring into action at any time.

As the show gained in popularity, celebrities began to get in on the act too. BattleBots was featured on an episode of the Tonight Show, where Jay Leno revealed a BattleBot of his own: Chin-Killa (**Figure 3**).

The show was also featured in an episode of the popular sitcom Malcolm in the Middle. Soon, a clothing line and remote controlled replicas of the robots were available for sale, introducing a new generation of viewers to robot combat. The show would go on to have a five season run, but the lasting legacy would endure, with BattleBots becoming the popular face of robot combat in America from that point on.

BattleBots Today

The end of the television series in 2002 was by no means the end of BattleBots. Since the show’s end, BattleBots has continued to host tournaments across the country. BotsIQ — a competition sponsored by BattleBots — focuses on teaching robotics, as well as other STEM areas to middle and high school aged students.

For information on upcoming events and how to get started in a BattleBots or BotsIQ competition, visit **BattleBots.com**.

Next up in our series: Post-BattleBots era competitions. SV

EVENT REPORT:

“Beware the Hobby that Eats” (Benjamin Franklin): Franklin Institute 2011

● by Pete Smith

The Franklin Institute in Philadelphia, PA (**www2.fi.edu**) was host to the Northeast Robotics Club’s (**www.nerc.us**) 5th Autumn Event last year on October 15th.

Forty-five Bots had entered, spread over the five different weight classes — from the 1 lb Ants up to the 30 lb Featherweight and Sportsman classes. With few no shows, it meant there would be a busy fight schedule.

Competitors started arriving just after 7:00 am, and fights started at 10:00 pm when the venue opened to the public. Pit space was tighter than last year with more space allocated to the audience, but this proved worthwhile with good crowds throughout the day. A new feature this year was a large flat screen TV facing the pits. This was much appreciated as it allowed even those busy repairing their bots to keep up with the action and reduced the press of competitors around the arena for the most exciting fights.

The competition saw some new and/or much improved bots. Most significant of these were the new 30 lb shell spinner Tetanus (**Figure 1**) replacing the now retired

FIGURE 1. Tetanus.

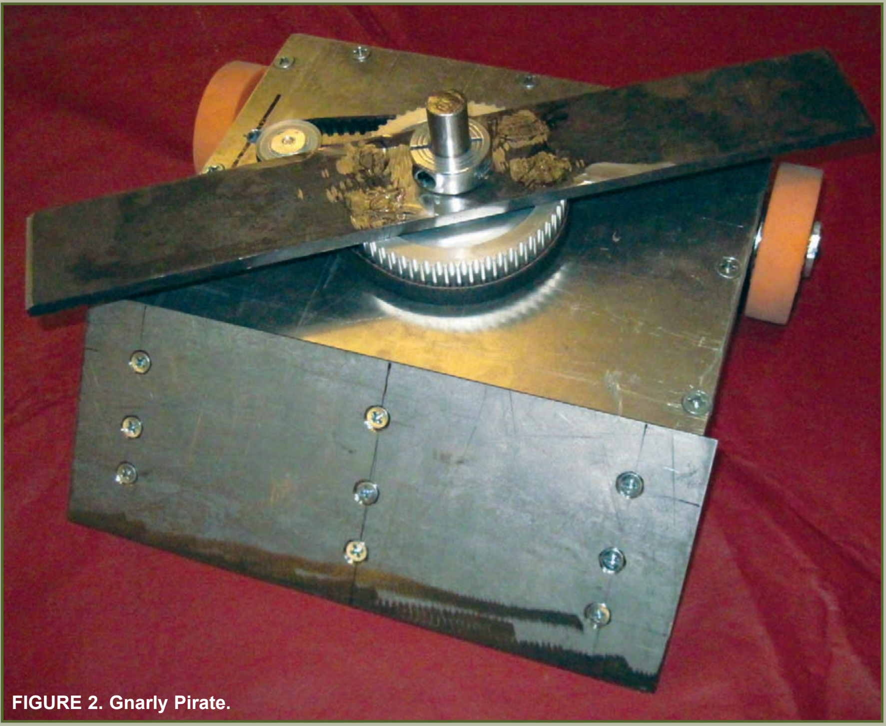


FIGURE 2. Gnarly Pirate.

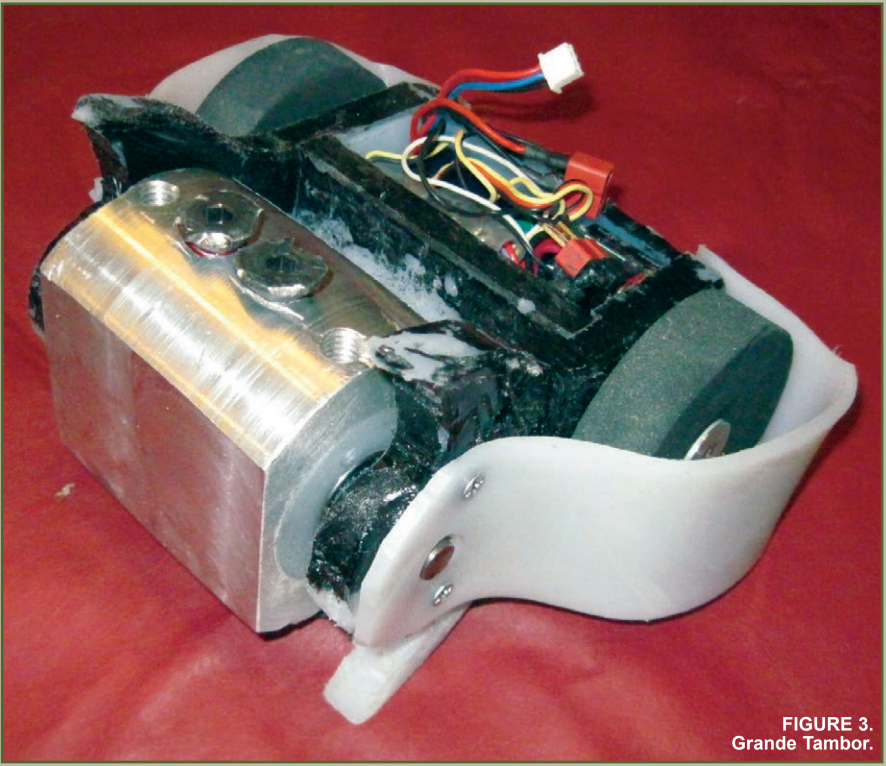


FIGURE 3. Grande Tambor.

Tripolar; a new 12 lb overhead bar spinner Gnarly Pirate (**Figure 2**); an improved Grande Tambor (**Figure 3**); and the first major competition for the new Kitbots (**www.kitbots.com**) 3 lb wedge kit Trilobite (**Figure 4**).

All the weight classes were the traditional double elimination, except for the Sportsman class which was round robin.

Nine 1 lb Antweights took part with brick Amatol and wedge Antelope vying with drum bot Poco Tambor for dominance, with Antelope taking the win in the final.

There was a strong field of 3 lb Beetleweights, including three Weta kits, a couple of BattleBots toy-based wedges, and a few other drumbots and wedges. The undoubted star was Grande Tambor. The weight of the massive drum had been reduced somewhat, but that weight had been put to good use to improve the wheel hubs and to add Weta style side armor. The bot could always mete out big hits but now it could also survive them itself.

Most opponents had some big air time before tapping out. Only Trilobite proved able to stand up to the drum. Its novel UHMW and titanium wedge design worked perfectly to prevent big hits, and its power was enough to get the best of any pushing matches. Trilobite won the first encounter early in the day and repeated the result in the finals to get first place.

The 12 lb Hobbyweights started with a bang with blade spinners Surgical Strike and Gnarly Pirate meeting in one of the first matches. Surgical Strike took the worst of the

RESULTS

	ANTWEIGHTS	BEETLEWEIGHTS	HOBBYWEIGHTS	SPORTSMAN	FEATHERWEIGHTS
1st:	Antelope	Trilobite	Fiasco	Enforcer	Mangi
2nd:	Poco Tambor	Grande Tambor	Surgical Strike	Phoenix	Tetanus
3rd:	Amatol	Play'n Krazy	Zandor	Shish Kabot	ODW

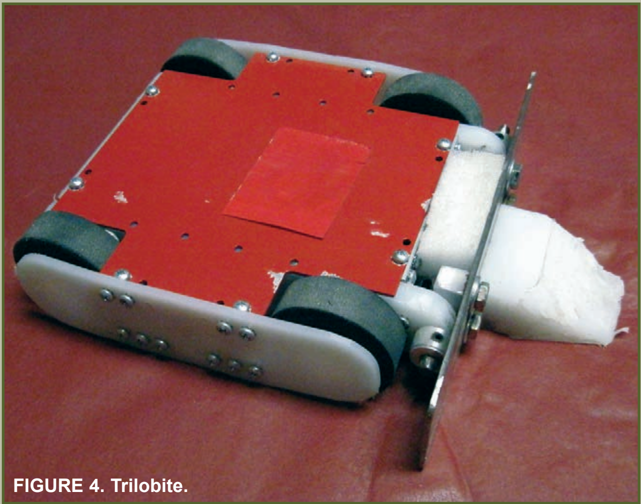


FIGURE 4. Trilobite.

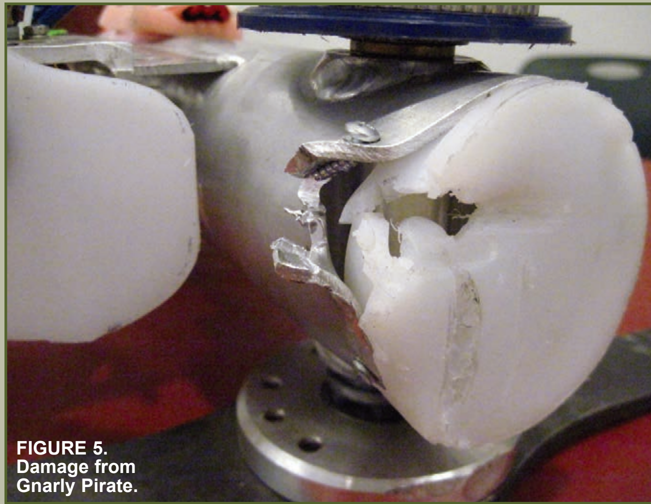


FIGURE 5. Damage from Gnarly Pirate.

damage (**Figure 5**) and was counted out. Fiasco and a repaired Surgical Strike (thanks to invaluable help from Chuck of Team Pneusance) ploughed through the winner's and loser's brackets until it met in a final that reduced both bots to near immobility as drives proved weaker than blades (**Figure 6**). Fiasco took a deserved first place in a judge's decision.

In the Sportsman class, the usually dominant flipper Upheaval was absent this year, so it gave some of the others their chance for fame. New flipper Phoenix and "blade on an arm" Enforcer were the most effective, with Enforcer going unbeaten for the win.

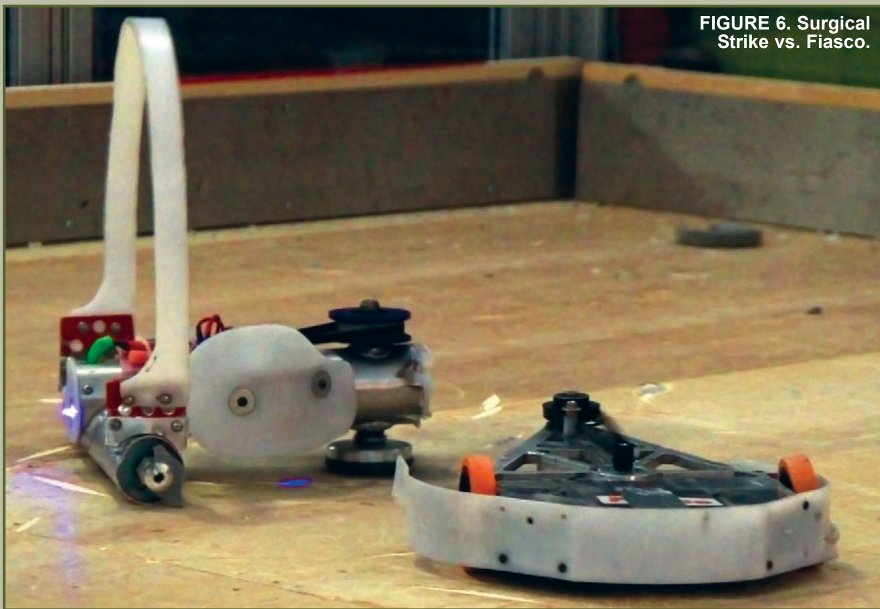


FIGURE 6. Surgical Strike vs. Fiasco.

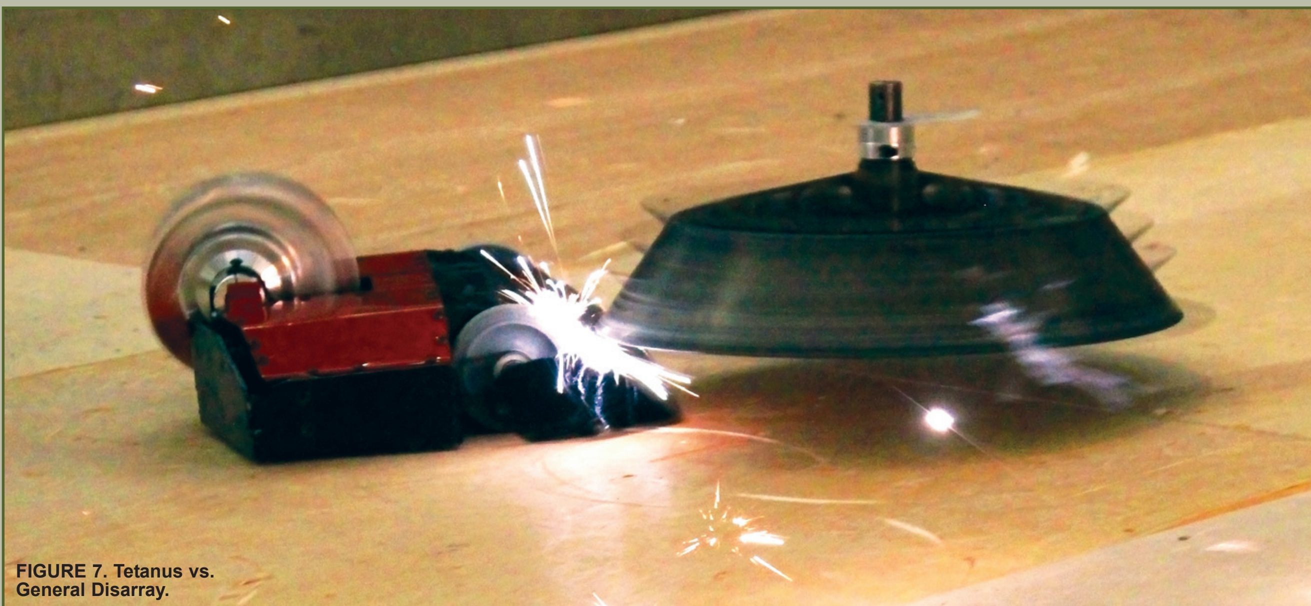


FIGURE 7. Tetanus vs. General Disarray.

FIGURE 8. Damage to Tetanus.

The new shell spinner Tetanus put on a great show in the 30 lb Featherweights division, including a tough match against lethal vertical spinner General Disarray (**Figure 7**). Tetanus won decisively after removing most of his opponent's armor, but at the cost of many of the teeth on his shell (**Figure 8**).

This may have cost him first place after facing the thick UHMW armor on the hammer bot Mangi in the finals.

The next major competition for NERC is scheduled to take place February 18–19, 2012 at Motorama in Harrisburg, PA.

For details check out **www.buildersdb.com**, the RFL **http://botleague.net**, and NERC at **www.nerc.us**. Come and watch or build a bot and compete — you will be very welcome!

You can see many of the fights mentioned in this article on YouTube by searching the bot's name and "Franklin." SV

*Photos by author and Brian Benson (**www.bensonpv.com**)*

EVENT REPORT:

PennBots — The Battle at Yellow Breeches

● by Dave Graham

PennBots — the Robot Club of Pennsylvania — hosted their Fall Fling on November 19, 2011 at the Yellow Breeches Middle School in Boiling Springs, PA. This was an important event for PennBots in that it was their first event advertised on **buildersdb.com**. That advertising paid off as fighting robot competitors (**Figure 1**) from Pennsylvania, New Jersey, and Virginia met to determine who was the best in three insect weight classes: Flea (a.k.a., Fairy) at 150 grams, Ant at one pound, and Beetle at three pounds. Corporate sponsors *SERVO Magazine*, Pololu, FingerTech Robotics, Grand Wing System (GWS), and the BotBrain Company also responded favorably to the advertising with generous donations of fighting robot materials for the winners.

The Flea competition was a little thin — originally, six competitors registered their Fleas, but only three toed the line on fight day. The fighting Fleas were my horizontal spinner Hedgehog (**Figure 2**), vertical spinner Tomahawk (**Figure 3**), and Richard Kelley's latest Flea creation Baby Slicer (**Figure 4**). Baby Slicer is built on a custom-made fiberglass shell, and features part of a hacksaw blade as the spinning weapon.

Hedgehog and Baby Slicer fought first. After a few weapon-to-weapon hits, Hedgehog cut a wire on Baby Slicer's weapon motor, and then proceeded to cut both tires off of the bot (**Figure 5**). Baby Slicer then fought Tomahawk, and pretty much the same thing happened. Baby Slicer's weapon motor quit working and Tomahawk sliced off one of Baby's tires. At that point, Baby Slicer tapped out. It was all over very quickly with first place going to Hedgehog, second to Tomahawk, and third to Baby Slicer.

The Ant competition proved more exciting with an awesome collection of destructive weapons including spinning blades and drums, a beater, and a flipper. The first round got off to a fast start with Russell Hubbard's under cutter

Stormageddon (**Figure 6**) taking on Sean McKeown's beater Gyroscopic (**Figure 7**). Both bots moved to the center of the arena where their weapons "kissed" a couple times, and then BOOM! The two bots flew to opposite ends of the arena and came to rest in the two arena pit hazards. Technically, it was a double knockout. The judges inspected damage to both bots and awarded the victory to Russell Hubbard after seeing that Stormageddon had ripped the beater off of Gyroscopic.

Also in first round action, Bill Bechtel's creative flipper bot CrocBot (**Figure 8**) took on Ken Brandon's spinning drum Buzz (**Figure 9**). CrocBot took a beating, had his flipper ripped off, and was way behind on points until Buzz drove into the arena pit hazard. CrocBot's victory earned him a second round match against Stormageddon, where he took a second beating and moved to the loser's bracket.

In second round action, my

FIGURE 1. PennBots Fall Fling competitors and their bots.

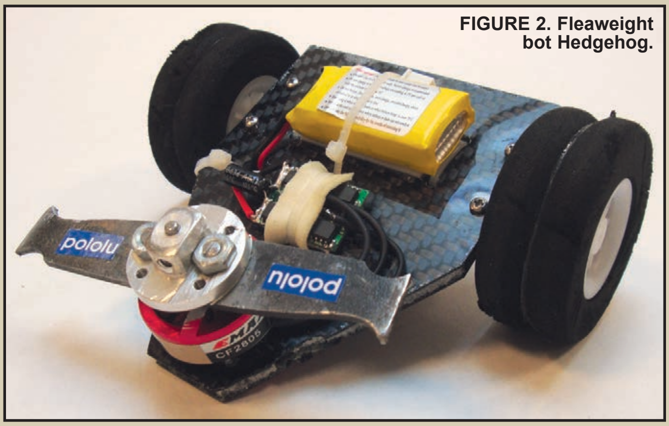


FIGURE 2. Fleaweight bot Hedgehog.

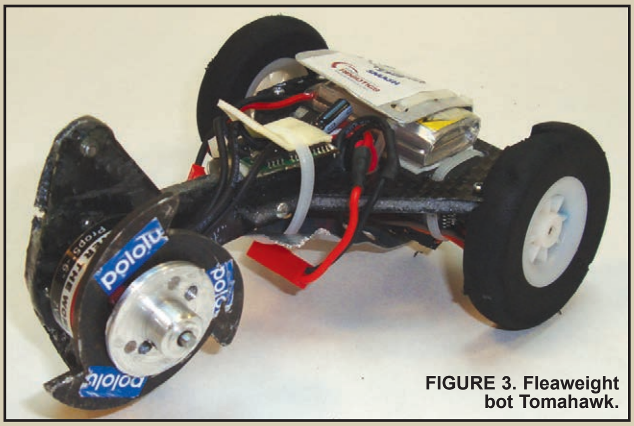


FIGURE 3. Fleaweight bot Tomahawk.

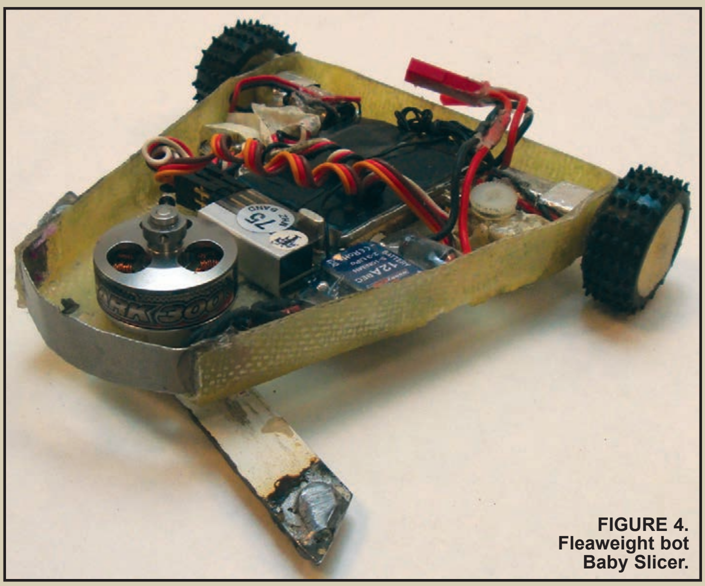


FIGURE 4. Fleaweight bot Baby Slicer.

FIGURE 5. Fleaweight bot Hedgehog cuts the tires off of Baby Slicer.

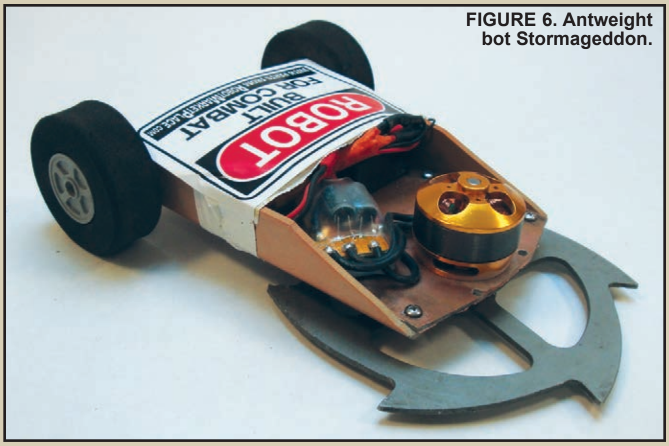


FIGURE 6. Antweight bot Stormageddon.

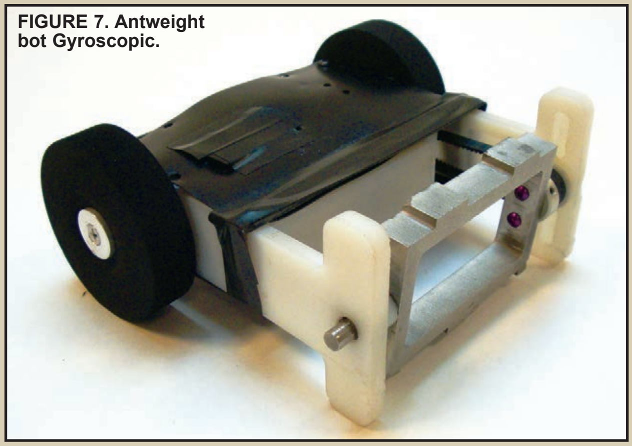


FIGURE 7. Antweight bot Gyroscopic.

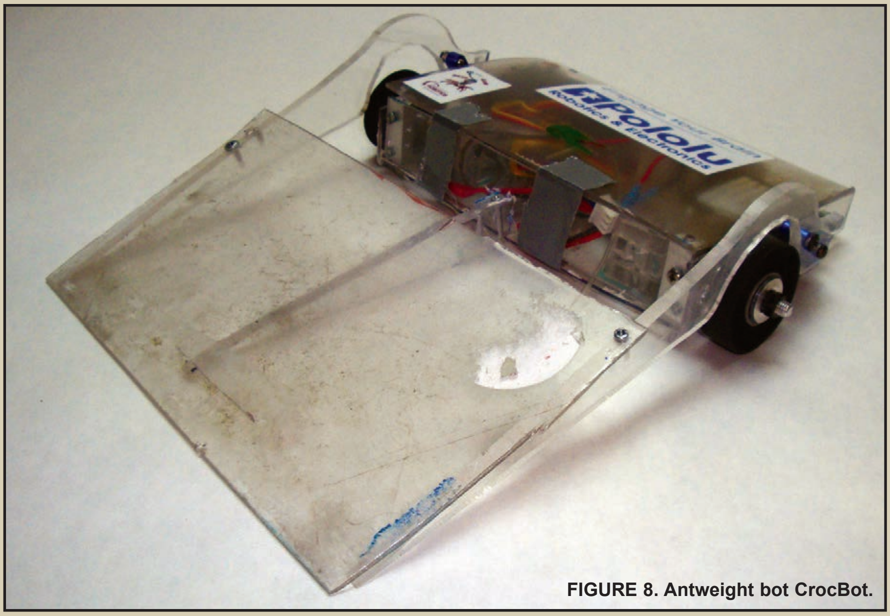


FIGURE 8. Antweight bot CrocBot.

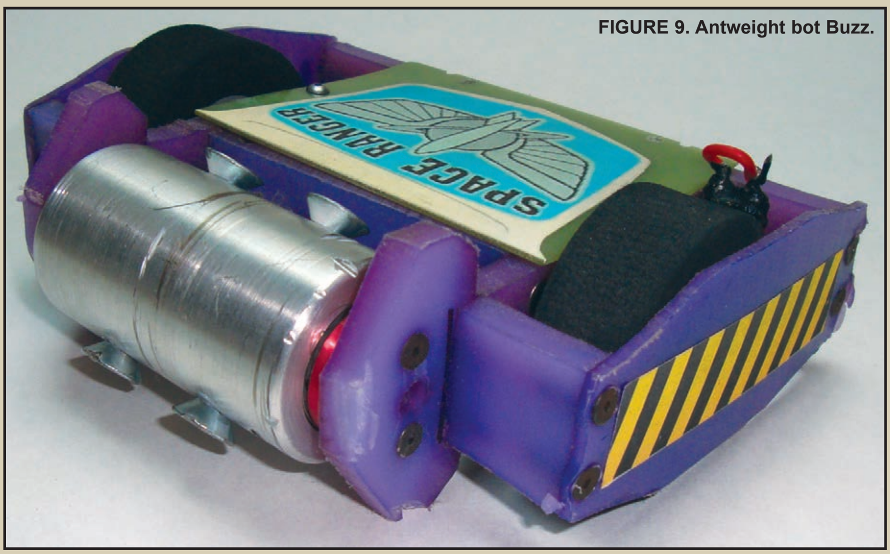


FIGURE 9. Antweight bot Buzz.

spinning blade bot Kyle's Cutter (**Figure 10**) took a bite out of the tire of Richard Kelley's latest Ant creation Slicer (**Figure 11**). Kyle's Cutter also fell prey to Stormageddon in the third round, suffering competition-ending damage when Stormageddon ripped the wheel and drive shaft out of the left drive motor (**Figure 12**); see the close-up in **Figure 13**. The win earned Stormageddon a spot in the Ant title match.

CrocBot and Buzz fought their way to the top of the loser's bracket and met a second time for a chance to fight Stormageddon in the Ant final match. This time, Buzz was victorious and prepared to do battle with Stormageddon for the Ant gold. Due to PennBots double elimination format, Buzz would have to beat Stormageddon twice in order to win. Buzz wonthe first match when Stormageddon drove into the arena pit hazard. In the second match, Buzz cornered Stormageddon and then forced him into the arena pit hazard again, giving him the second well-deserved victory and first place in the Antweight class.

The Beetleweight class had five competitors, however, two of the five bots were really just heavy Ants, weighing in at less than 1.5 pounds (they probably should have gotten together and fought as a multibot). In first round action, Richard Kelley's latest Beetle creation the Box (**Figure 14**) quickly disposed of Tim Thompson's spinner Thingamajigger (**Figure 15**). In the upset of the day, Sean McKeown and his lightweight bot Beetle (**Figure 16**) pushed Bill Bechtel's Beetle bully Topsy-Turvey (**Figure 17**) into the arena pit hazard (**Figure 18**).

TABLE 1 - WINNERS

	FLEA	ANT	BEETLE
1st:	Hedgehog *Dave Graham*	Buzz *Ken Brandon*	The Box *Richard Kelley*
2nd:	Tomahawk *Dave Graham*	Stormageddon *Russell Hubbard*	Topsy-Turvey *Bill Bechtel*
3rd:	Baby Slicer *Richard Kelley*	CrocBot *Bill Bechtel*	Beetle *Sean McKeown*
Melee:		Gyroscopic *Sean McKeown*	Topsy-Turvey *Bill Bechtel*

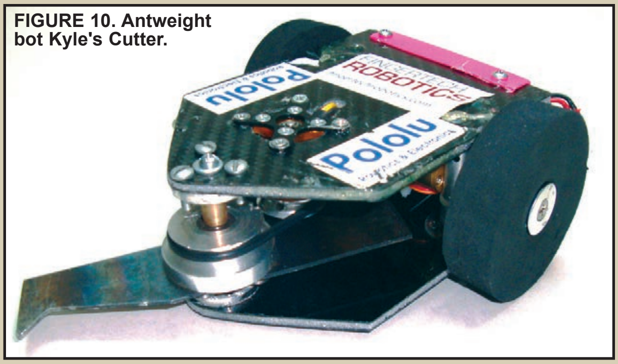


FIGURE 10. Antweight bot Kyle's Cutter.

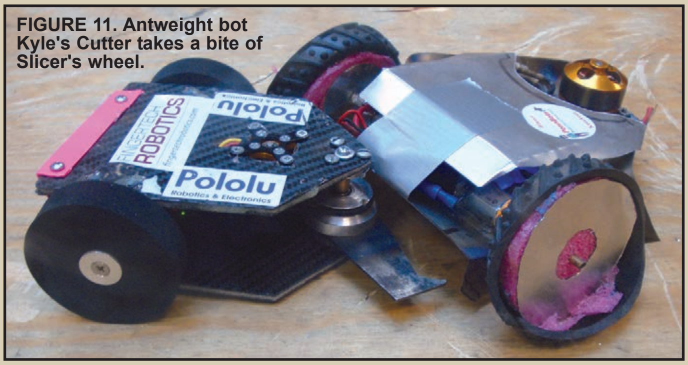


FIGURE 11. Antweight bot Kyle's Cutter takes a bite of Slicer's wheel.

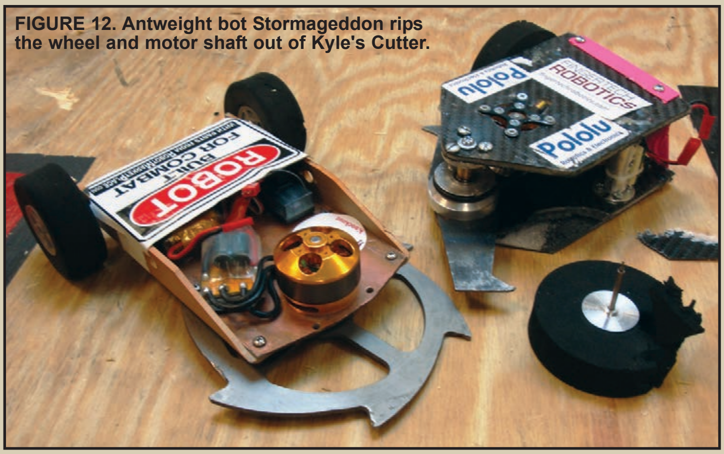


FIGURE 12. Antweight bot Stormageddon rips the wheel and motor shaft out of Kyle's Cutter.

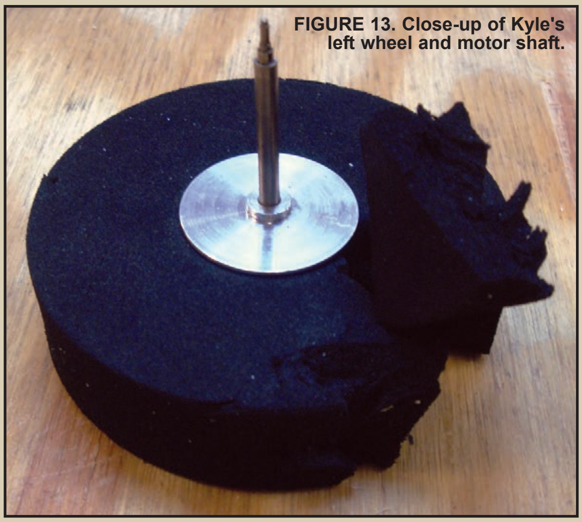


FIGURE 13. Close-up of Kyle's left wheel and motor shaft.

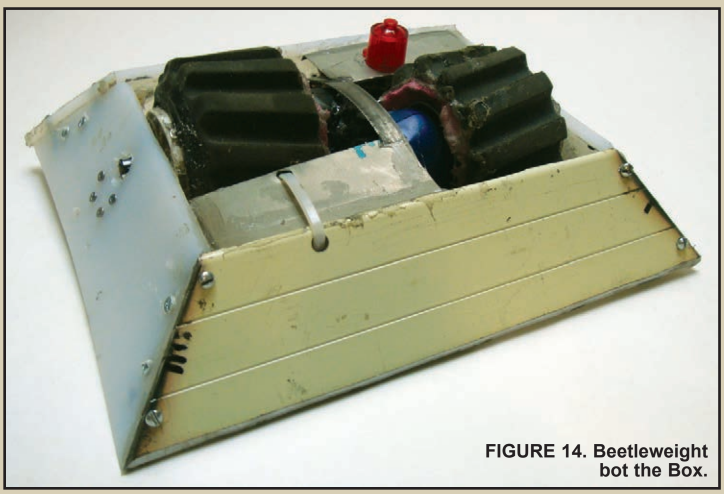


FIGURE 14. Beetleweight bot the Box.

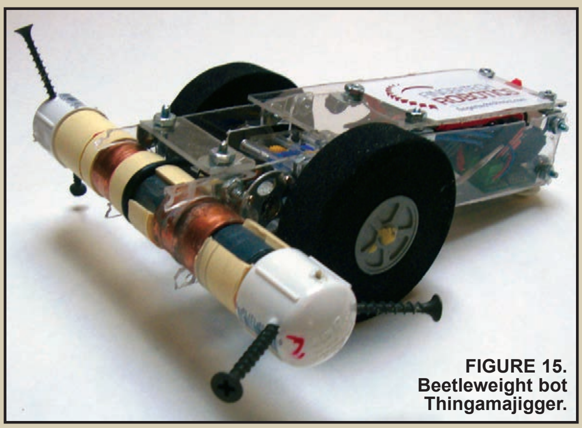


FIGURE 15. Beetleweight bot Thingamajigger.

FIGURE 16. Beetleweight bot Beetle.


FIGURE 17. Beetleweight bot Topsy-Turvey.


FIGURE 18. Beetleweight bot Beetle pushing Topsy-Turvey into the arena pit hazard.

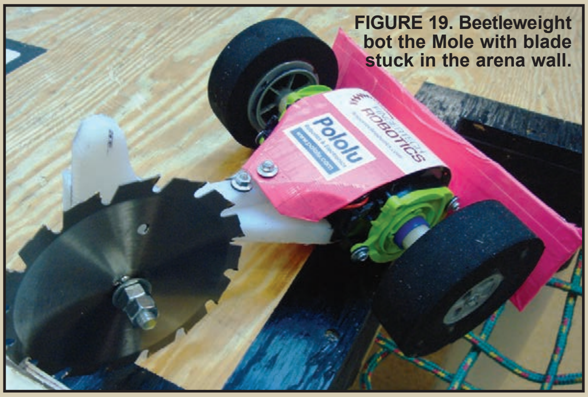


FIGURE 19. Beetleweight bot the Mole with blade stuck in the arena wall.

FIGURE 20. Antweight bots Buzz, Slicer, and Snaggletooth in the arena pit hazard.

FIGURE 21. Max (left), Evan (center), and dad Bill Bechtel (right).

I lost two straight matches with my rebuilt spinner the Mole. The best I could do was to stick myself on the arena wall (**Figure 19**). The Box beat Topsy-Turvey for the Beetle gold. A complete list of all the winners is in **Table 1**.

Melees were held for the Ants and Beetles, with the arena pit hazards coming out as the big winner since it was full of Ants Buzz, Slicer, and Snaggletooth (**Figure 20**).

Following the competition, the youngest competitors Max and Evan Bechtel posed with their dad, Bill — a Yellow Breeches Middle School science teacher and one of the original founders of PennBots (**Figure 21**). The Bechtel brothers then decided to enter the arena in a proof of concept test for their proposed 75 pound fighting weight class build, Bratbots (**Figure 22**).

Mark your calendars now for the next PennBots event: the 2nd Annual Downtown Dogfight on Saturday, March 24, 2012 in Harrisburg, PA. This event is a collaborative venture between PennBots and Harrisburg University, and will feature Insect class fighting robots and thinking robot competitions including line following and maze solving.

Last year's inaugural event had great corporate sponsorship and television coverage from two local stations. This event continues to grow, so register early and plan to attend this great day of robotmania! You can follow PennBots on their website at **www.pennbots.org** or on **www.buildersdb.com**. SV

FIGURE 22. Max (left) and Evan (right) Bechtel in the PennBots arena.

EVENTS:

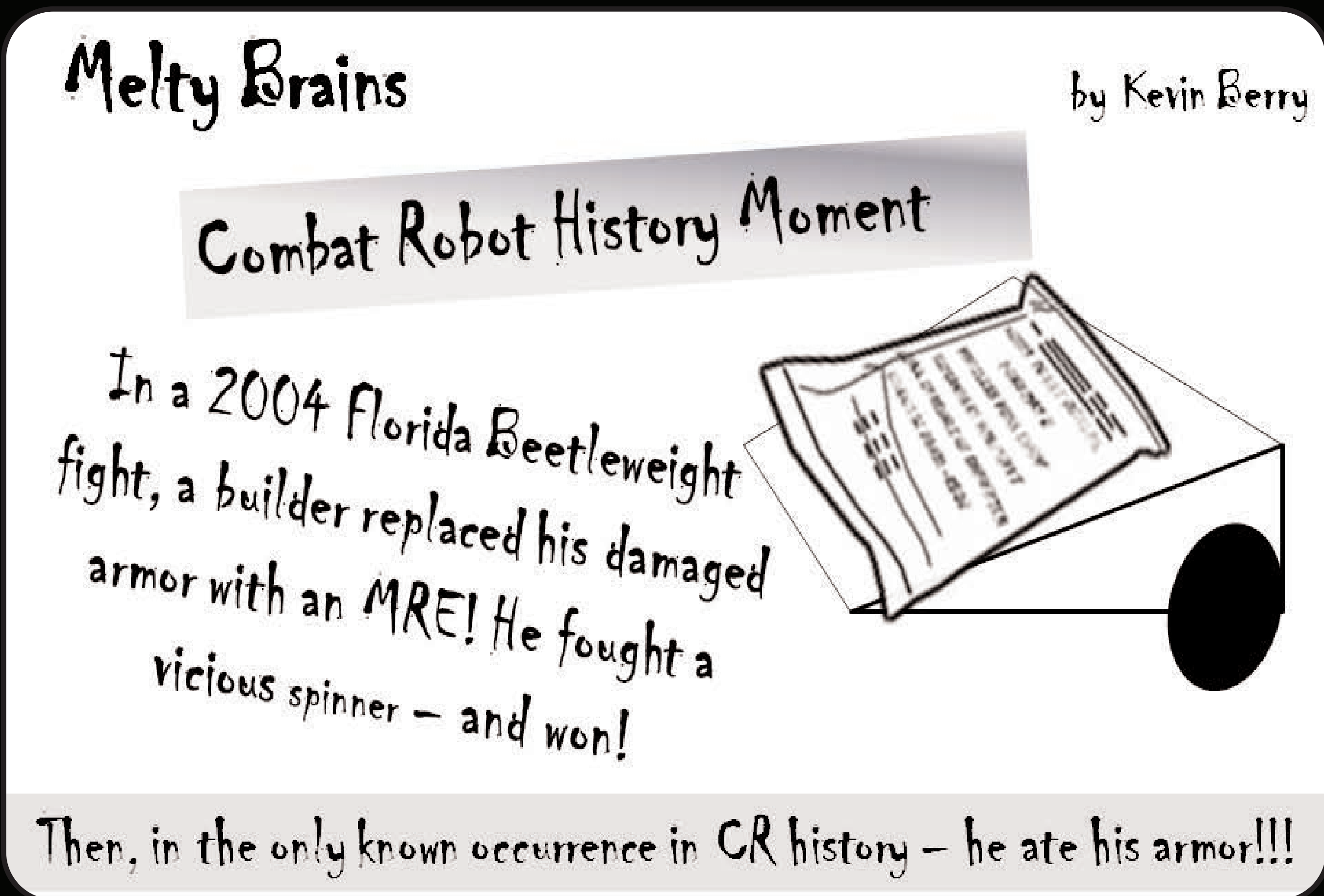
Upcoming Events for February – March 2012

NWMHobby Expo 2012 will be presented by Western Allied Robotics in Monroe, WA on February 11th. For more information, go to **www.westernalliedrobotics.com**.

Motorama 2012 will be presented by the North East Robotics Club in Harrisburg, PA February 17–19, 2012. For more information, go to **www.nerc.us**.

The Central Illinois Bot Brawl 2012 will be presented by the Central Illinois Robotics Club in Peoria, IL on March 24th. For more information, go to **http://circ.mtco.com**.

The 2nd Annual Downtown Dogfight will be presented by PennBots at Harrisburg University in Harrisburg, PA on Saturday, March 24, 2012. For more information, go to **www.pennbots.org**. **SV**



A Peek Inside the NEATO XV-11

by Bryan Bergeron

www.servomagazine.com/index.php?/magazine/article/february2012_Bergeron

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

FIGURE 1. NEATO XV-11 and wall charger unit.

Of the practical applications of robotics in the home, the robot vacuum cleaner is the most prominent. Whisking away the crumbs of daily activity certainly qualifies for at least two of the tasks robots are good at – the dull and dirty. Join me for a teardown of the NEATO XV-11 home vacuum cleaner, from NEATO Robotics (*www.neatorobotics.com*) of Palo Alto, CA. The robot and wall charger unit are shown in Figure 1.

Overview

As home robot vacuum cleaners go, the NEATO is priced in the upper range of automated cleaners from iRobot and others. Amazon lists the XV-11 at $390. Although the folks at NEATO would love for me to comment on the relative efficacy of the NEATO as a room vacuum cleaner, I won't. If you're an avid *SERVO* reader, you're not looking at this article for advice on which robot vac is best at picking up cat hair, but which would make the best platform for experimentation and for harvesting parts.

The NEATO doesn't disappoint on these latter two categories. The *piece de resistance* of the NEATO is a light radar or LIDAR with full 360 degree coverage. In comparison, several years ago I paid nearly $2,500 for a Hokuyo URG-04LX LIDAR for my rover, and it only provides 240 degree coverage with a range of 12 feet. More on the LIDAR later – let's get to the teardown.

Teardown

If you've read my book on *Teardowns* [1], you know that I give the physical layout and construction of a device as much emphasis as the underlying electronic design. I've found everything from trails of hot glue, cold solder joints, and bits of waste plastic, to loose screws and hair in teardowns. These and similar findings point to cost-cutting maneuvers by the manufacturer, and suggest the electronic components aren't first rate. However, that's one of the

exciting things about teardowns – you never know what you'll find behind a stylish cover.

The NEATO ships with two components: a wall charger and the robot. Let's start with the charger; shown in use in **Figure 1** and with the cover removed in **Figure 2**. Externally, the charger unit is designed to reflect the IR beam from the LIDAR in a way that enables the robot to locate the charging station when the charge of the batteries is low. You can see the light-dark-light bands of the reflective surface in **Figure 1**.

As shown in **Figure 2**, the charger is a typical 'brick' found with many laptop computers. It provides 24 VDC at 2.5A, through either a direct cable connection or the metal contacts on the back of the robot, shown in **Figure 3**. The unit has a clean, simple layout with lots of air for heat dissipation.

The NEATO tears down quickly and easily with only a Philips screw driver. There are no hidden springs or screws to impede your progress. Before diving in, be sure to note the location of the USB and power ports in **Figure 3**, the IR rangefinder sensors on the sides of the unit (**Figure 4**), and the all-important LIDAR unit (**Figure 5**). **Figure 6** shows the ample cargo space available with the dust collection bin removed.

Flip the unit over (**Figure 7**) and note how the wheels pop out. Sensors attached to the wheel assemblies detect this condition and the CPU issues a "Please Put Me Down" message on the LCD when this occurs. Also note the pair of passive wheels in the rounded rear of the unit. The flat front of the unit is spring-loaded, with sensors to detect collisions. Remove the two plastic retaining plates to reveal the sizeable NiMH battery packs shown in **Figure 8**. The 7.2V packs are rated at 3,200 mAh.

Next, remove the remaining dozen or so screws visible on the

FIGURE 2. Wall charger unit with cover removed.

FIGURE 3. Back of the robot, showing the pair of wall charger contacts (center). Note the USB and power ports to the left, below the LCD screen.

FIGURE 4. An IR rangefinder sensor on the side of the robot.

FIGURE 5. The LIDAR unit. Note the plastic supports for the cover.

FIGURE 6. Cargo space with dust collection bin removed.

FIGURE 7. Underside of robot.

bottom of the unit and remove the plastic cover to reveal the main blower and drive assembly as shown in **Figure 9**. The layout is clean and balanced. Remove the 12 VDC at 3.8A blower by unplugging the cable and sliding the blower from the friction mount pegs.

Remove the two drive assemblies by first sliding the metal pin out from the pivot point and then unhooking the single spring as in **Figure 10**. Unplug the cable and set the assemblies aside. Remove the brush assembly by lifting one edge to unhook it from the main body. Set the brush aside. Now, flip the robot over so that the LIDAR is visible and separate the halves of the clamshell, revealing the LIDAR unit, the main circuit board, and the brush motor as in **Figure 11**. You'll have to disconnect a ribbon cable and a couple of power cables to completely remove the cover which contains the LCD screen and associated circuitry.

Remove the LIDAR unit by releasing four screws, visible in **Figure 12**. Note the belt drive. While you have the LIDAR in your hands, you might as well remove the four screws in the cover to reveal the laser diode and receiver shown in **Figure 13**. Note that the entire transmitter/receiver assembly rotates.

Now, let's look at the other sensors exposed in the robot. **Figure 14** shows a top-down view of two bumper sensors. Remove the single screw holding the metal cover in place to reveal an IR rangefinder sensor and magnetic sensor as in **Figure 15**. Note the connector on the rear of the IR rangefinder in the figure. The magnetic sensor is used to detect strips of magnetic tape that you can place across doorways and other areas that you don't want disturbed. **Figure 16** shows the other corner of the robot with the same sensor array, as well as the 12 VDC sweeper motor.

FIGURE 8. The 7.2V at 3,200 mAh NiMH battery packs.

Finally, **Figure 17** shows the circuit board with the ATMEL ARM processor taking center stage. The layout is clean, solder joints bright, and everything is connected by cables with workable connectors. In other words, if you plan to swap out sensors or monitor signals with a second processor, you won't have to pull out your soldering iron.

All in all, a pleasant teardown. The NEATO is an example of a well-engineered device that lends itself to repair and modification. Just what the robotics enthusiast ordered.

LIDAR Details

According to a paper published by the IEEE [2], the LIDAR used on the NEATO is both impressive functionally and affordable. Obviously, the LIDAR is intended to operate indoors – even an IR laser would be washed out by direct sunlight. Furthermore, put out on an open patio without four walls, the LIDAR wouldn't receive the reflections the onboard CPU needs to define the space to traverse.

Range is listed at six meters, with an angular resolution of one degree. At 10 revolutions per second, the LIDAR makes 4,000 readings per second. Power consumption is a modest two watts. Unlike my much more expensive Hokuyo, there is no spinning mirror involved. Instead, the entire optical assembly rotates. In addition, unlike the Hokuyo — which measures the time required for light to travel from the laser diode to a structure or object and back again — the NEATO LIDAR uses simple triangulation.

The downside of triangulation is that accuracy decreases with distance from the unit. This is because transmitted and reflected beams are essentially parallel with targets from five or six meters to infinity. The engineers at NEATO

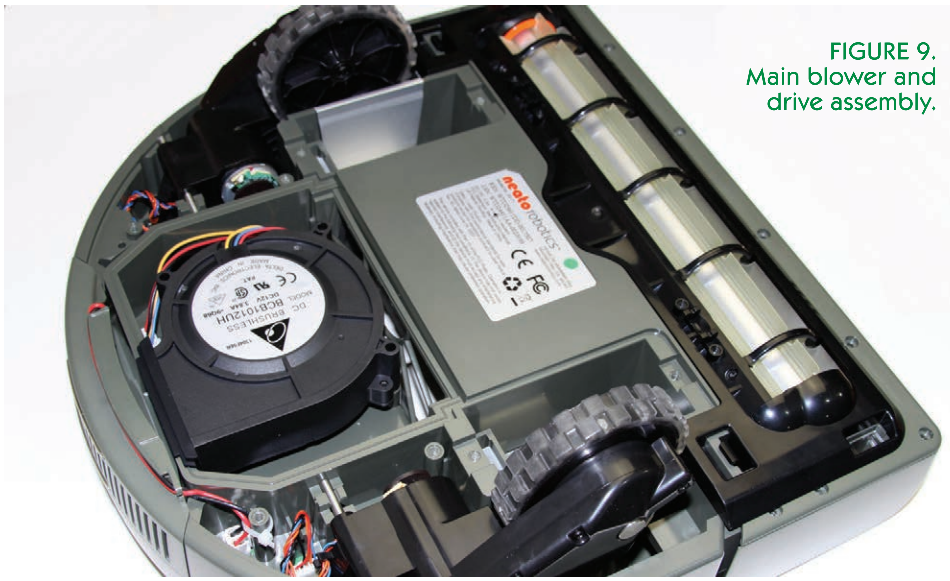


FIGURE 9. Main blower and drive assembly.

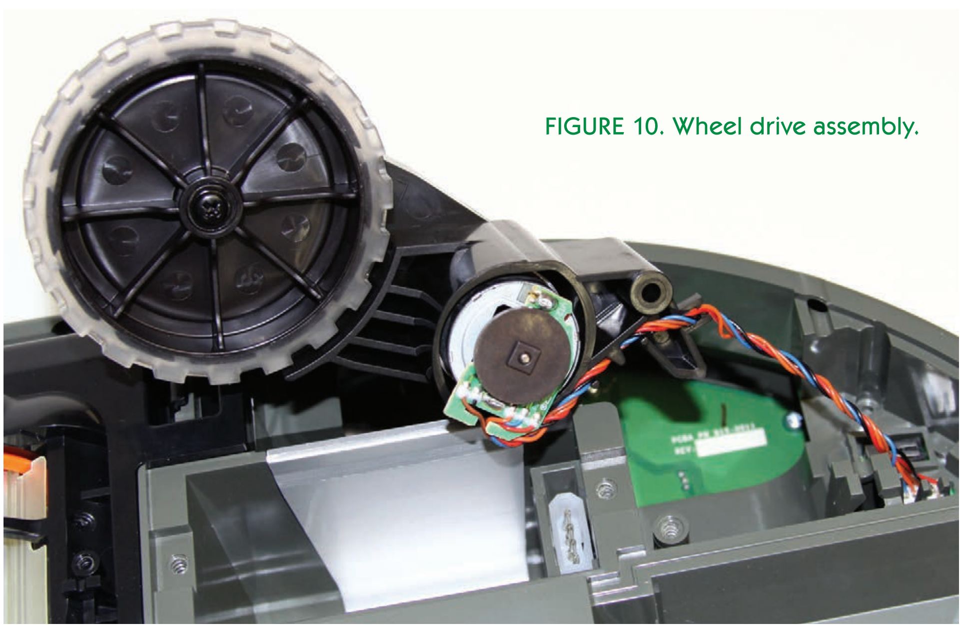


FIGURE 10. Wheel drive assembly.

FIGURE 11. Clamshell cover removed revealing LIDAR and circuit board.

addressed the accuracy issue by painting whatever is in line with the LIDAR with dots – that is, by pulsing the laser – and then determining the center of each dot. You can see this 'dot spraying' by turning down the room lights and recording the NEATO in operation with your video camera. Most video cameras are more sensitive to IR than is the human eye. The LIDAR also maximizes accuracy by limiting extraneous light with light filters.

The USB Port and ROS

This teardown is primarily about hardware, but I couldn't resist accessing the NEATO's Robot Operating System (ROS) through the USB port with a type A to mini-B cable. You can also communicate with the NEATO via Bluetooth. In short, you can issue commands to run the motors and read the sensors without cracking the case. A built-in help function is available. To illustrate, let's set a motor to run in a particular vector with the SetMotor command. The format of the command is:

```
SetMotor
[LWheelDist <LWheelDist_value>]
[RWheelDist <RWheelDist_value>]
[Speed <Speed_value>]
[Accel <Accel_value>]
[RPM <RPM_value>]
[Brush]
[VacuumOn]
[VacuumOff]
[VacuumSpeed <VacuumSpeed_value>]
[RWheelDisable]
[LWheelDisable]
[BrushDisable]
[RWheelEnable]
[LWheelEnable]
[BrushEnable]
```

As you might expect, ***Accel*** is the desired acceleration in millimeters/second/second, and ***VacuumOFF*** turns off the vacuum motor.

Once you've checked out the built-in

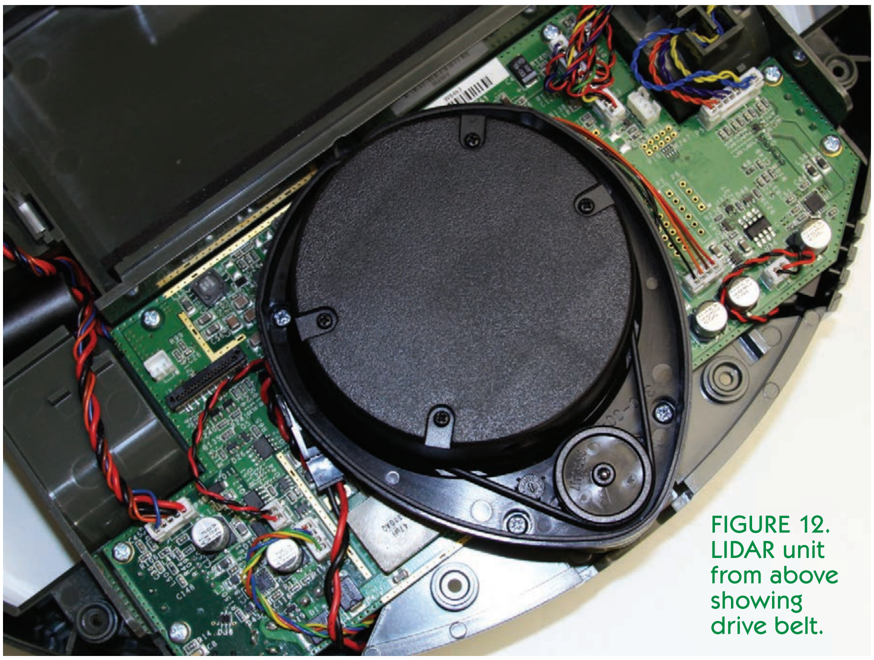
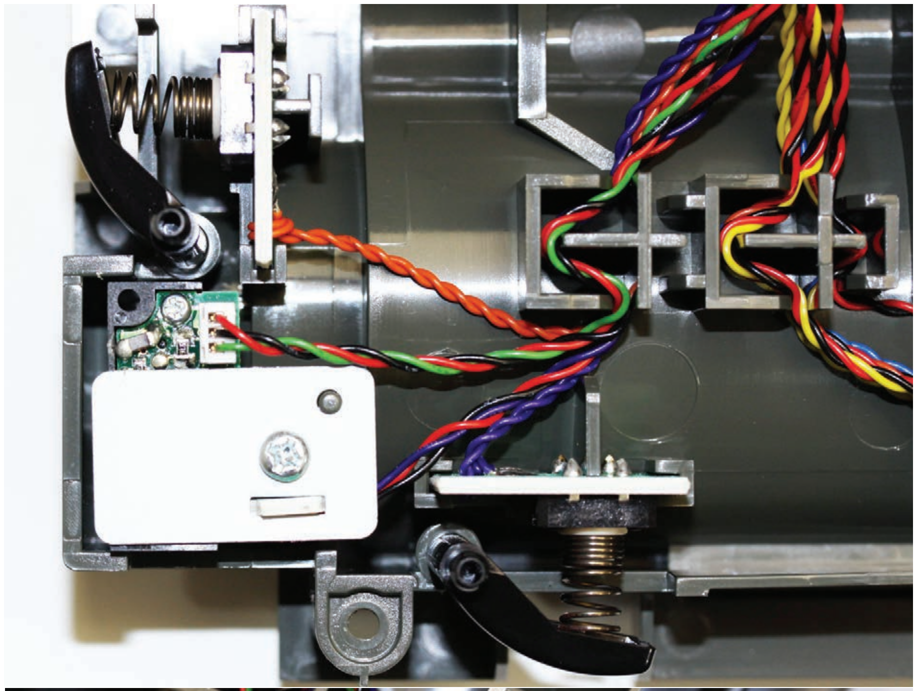


FIGURE 12. LIDAR unit from above showing drive belt.

FIGURE 13. LIDAR unit with rotating IR laser and receiver.

FIGURE 14. Corner sensor assembly.



functions and commands, you can leverage the work of others to get at the underlying operating system. Check out the hacks available for the NEATO at Hobby Robotics (**http://hbrobotics.org/wiki/index.php?title=Dave%27s_XV-11_notes**). You'll need the ROS driver for the NEATO from Willow Garage (**www.ros.org/wiki/neato_robot**). The documentation isn't great, but there are samples on how to handle the LIDAR scans and odometry data.

Platform Ideas

As illustrated in the teardown, the NEATO has considerable payload space for additional batteries, processor boards, sensors, and actuators. The fan, sweeper and sweeper motor, dust collection bin, and associated filter can be removed to minimize weight and battery drain. You can either drill out or replace the front plastic grill piece to make room for a small rear-facing video camera.

I haven't tried running the NEATO with the sweeper fan removed; worst case, you might have to use a resistor to fool any routine within the CPU that detects motor failure.

I'm working on mounting my Lynxmotion arm atop the NEATO. The obvious challenge with mounting anything above the case is potential interference with the LIDAR. However, if you look closely at the photos, you can see that coverage isn't really 360 degrees; it's broken by plastic supports – two front-facing and three rear-facing.

Although the fit is tight on the side of the LIDAR near the control panel, there's space for stainless steel posts in line with each plastic support. Attach an aluminum disc to the supports and you have a sturdy platform for an arm, swivel mount for a camera, or other small device. You could also mount a small video camera with hot glue

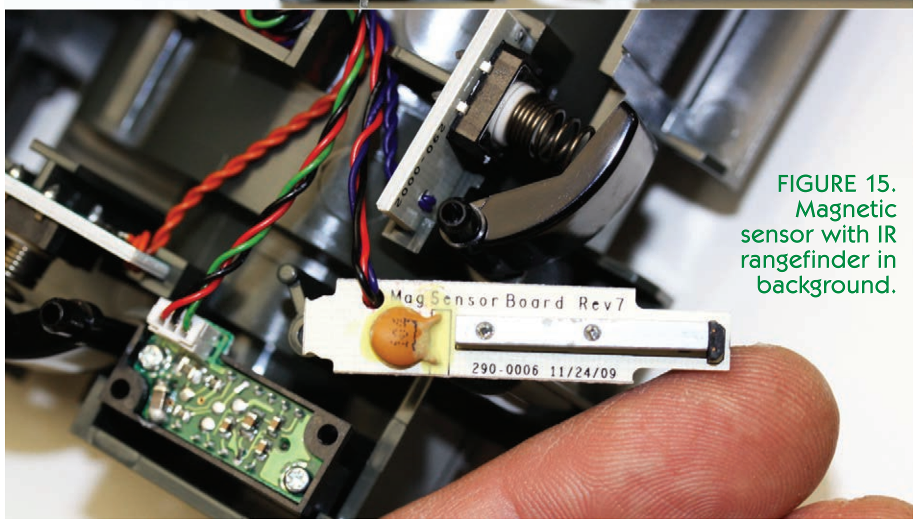


FIGURE 15. Magnetic sensor with IR rangefinder in background.

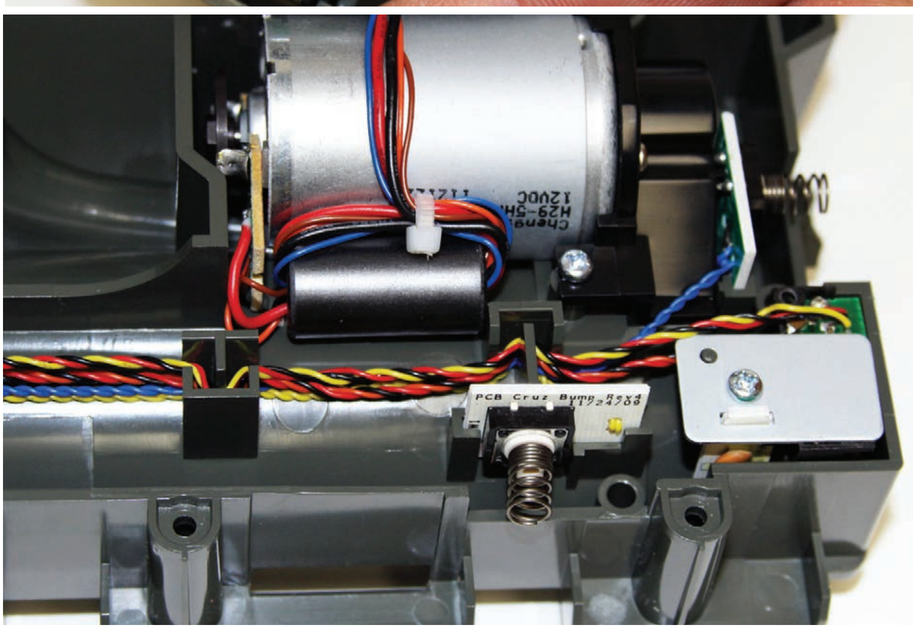


FIGURE 16. Brush drive motor.

directly to the top of the plastic LIDAR cover, and run the cables in line with the plastic supports.

Of course, the most obvious platform is to use the NEATO hardware as-is, and to make changes in the onboard firmware. The USB port provides access to the processor, but at this time, the unit's firmware and OS haven't been fully hacked. I expect this to change shortly. According to the folks at NEATO, parts will be available soon — at reasonable prices — and that's a start.

More Info

I'd like to thank Camp Peavy at Neato Robotics for the teardown specimen and for providing information on the NEATO. Although schematics and parts aren't available at this time, Neato Robotics is in the process of setting up service centers.

For third-party information on the NEATO, check out the write-up at **www.sparkfun.com**. The SparkFun folks have some preliminary data on the LIDAR. Details on the theory of operation of the LIDAR are at **www.robotshop.com/content/PDF/revolds-whitepaper.pdf**. The video at **www.youtube.com/watch?v=bJVEFlbuFO4** is worth watching – if only for inspiration on how to mount a laptop to the LIDAR top with a few strips of Velcro. It's only a matter of time before someone writes an iPad app to control the NEATO.

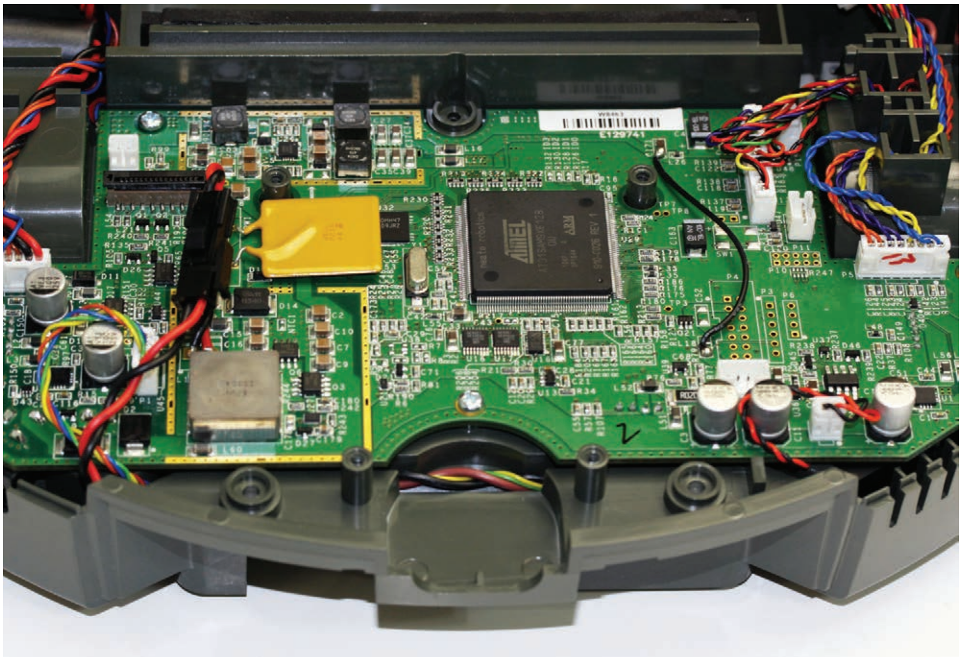


FIGURE 17. Circuit board with ARM processor.

Summary

As a cleaning appliance, the NEATO is well built, sturdy, and — at least in my opinion — underemployed. Although not marketed as an educational or hobby robot, it has a lot going for it. It's an excellent platform to investigate an open source robot operating system that's configured to operate in the real world. With the robot operating system driver, you can get laser scans, create maps, and navigate to waypoints. And if you don't really need your floors cleaned, the ample payload space can hold a variety of sensors and processors – or even a sandwich. **SV**

References

[1] Bergeron, B. *Teardowns: Learn how things work by taking them apart.* McGraw-Hill. 2010.

[2] Konolige, K, et. al. *A Low-Cost Laser Distance Sensor.* 2008 IEEE Intl Conference on Robotics and Automation. Pasadena, CA. p. 3002-3008.

Building The Pathfinder Rover

by Bogdan Chifor

www.servomagazine.com/index.php?/magazine/article/february2012_Chifor

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

The Military Technical Academy's contest entry for the recent Digilent Design Contest — the Pathfinder Rover — was awarded first prize in the European Finals and second prize in the Worldwide Finals. This rover — the work of two students: Bogdan Chifor and Nicolae Rosia — explores the idea of creating a versatile, minimalist, and low cost platform that could replace humans in hostile surveillance and exploration missions.

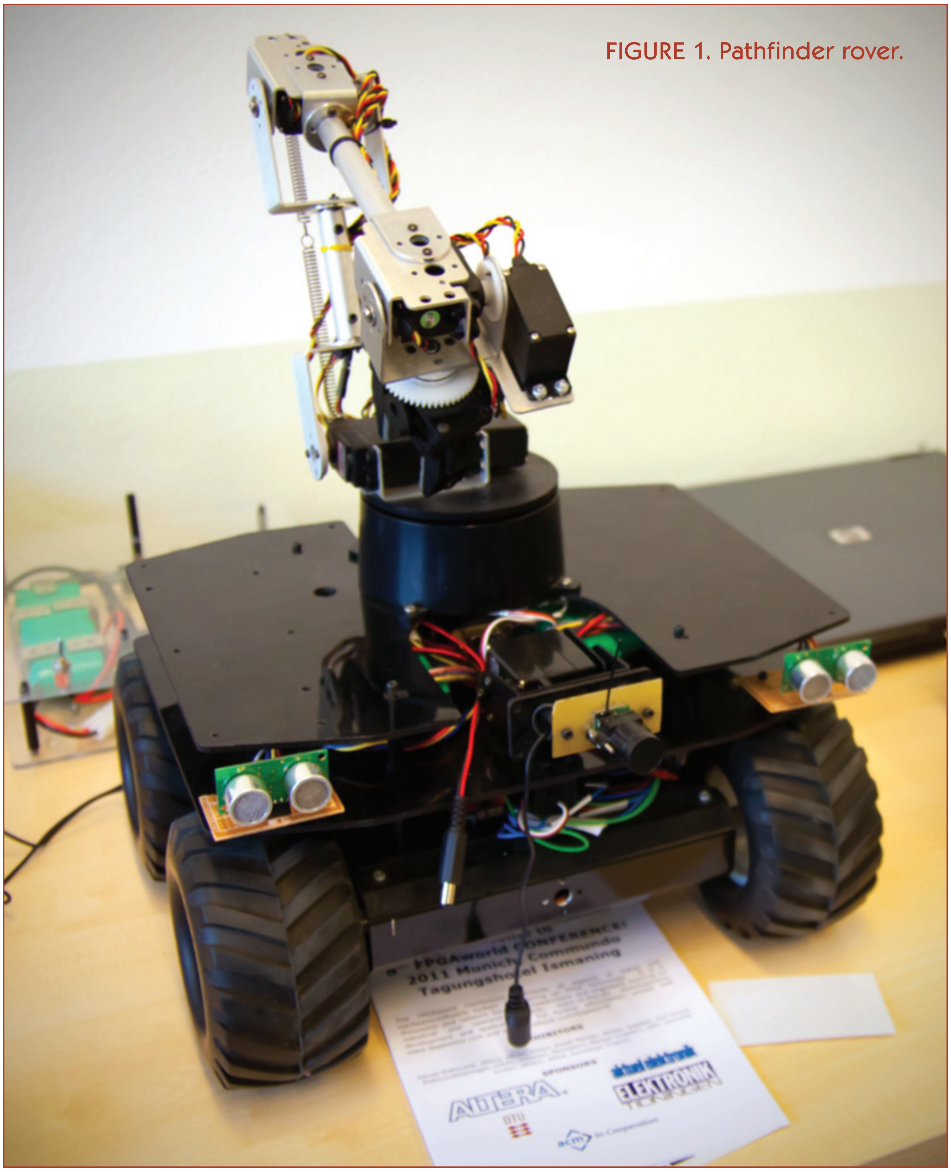


FIGURE 1. Pathfinder rover.

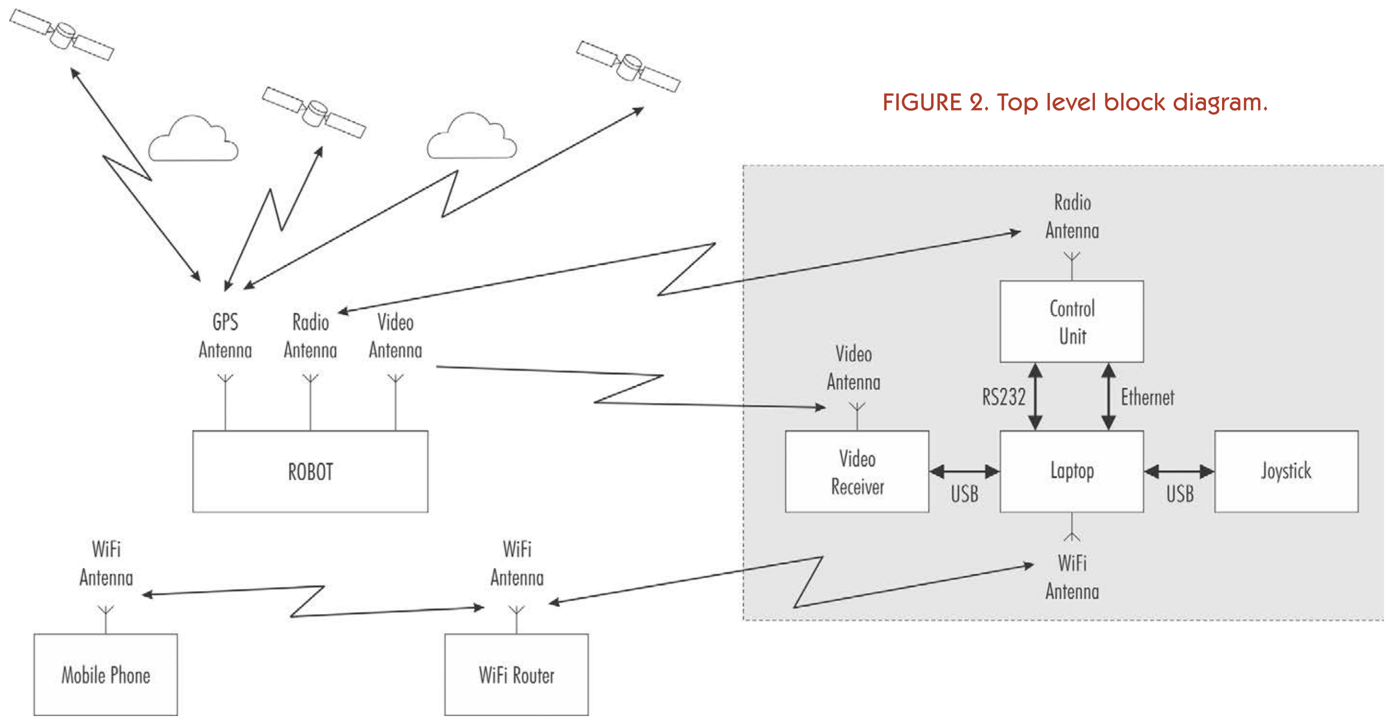


FIGURE 2. Top level block diagram.

Features In Brief

The Pathfinder rover presents a series of features that help the operator accomplish a remote surveillance mission. The system's main features are its robotic arm and onboard camera fitted on a pan/tilt mechanism, making it possible to gather all the information from the area of interest, and the capability to respond in real time to the operator's requests (the latter being limited by the radio link's coverage).

It can operate in two modes: manual and autonomous. In manual mode, the operator controls the rover using the joystick; while in autonomous mode, the robot follows given waypoints. In both of these modes, the robot's onboard MCU processes the information coming from the GPS, accelerometer, gyroscope, magnetometer sensors, and the wheel's encoders making both local and global (when GPS fix is available) localization information available to the operator. It also allows the manipulation of the pan/tilt mechanism on which the wireless onboard camera is mounted and of the fitted robotic arm.

Why This Project?

First of all, our aim was to learn something new and have fun doing it. So, what could be better than a project that has both an educational and practical value? This project was our chance to explore artificial intelligence, improve our skills in electronics (since we are both studying Computer Science), learn a new MCU architecture (PIC32MX), and use a radio transceiver. Also, we had the opportunity to apply what we learned in Control Theory, get familiar with tank-style kinematics, accelerometers, gyroscopes, magnetometers, and — last but not least — use our knowledge of Qt framework to design a user-friendly GUI (Graphical User Interface).

Design Overview

Our project consists of two major components: the

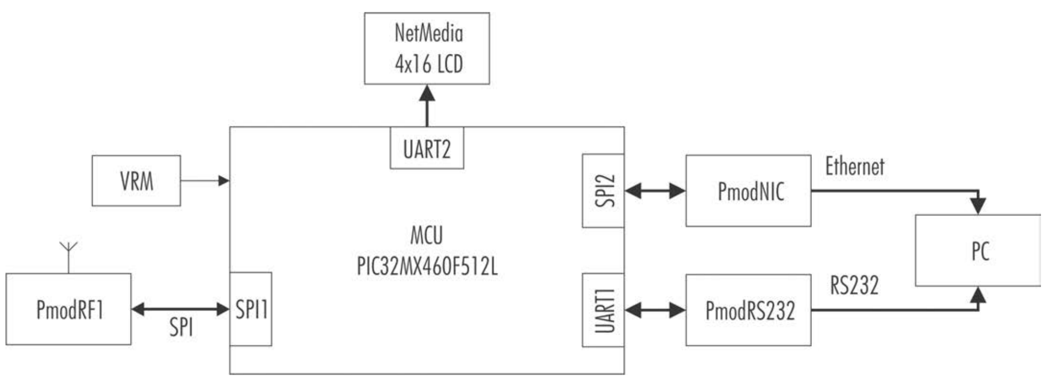


FIGURE 3. Control unit block diagram.

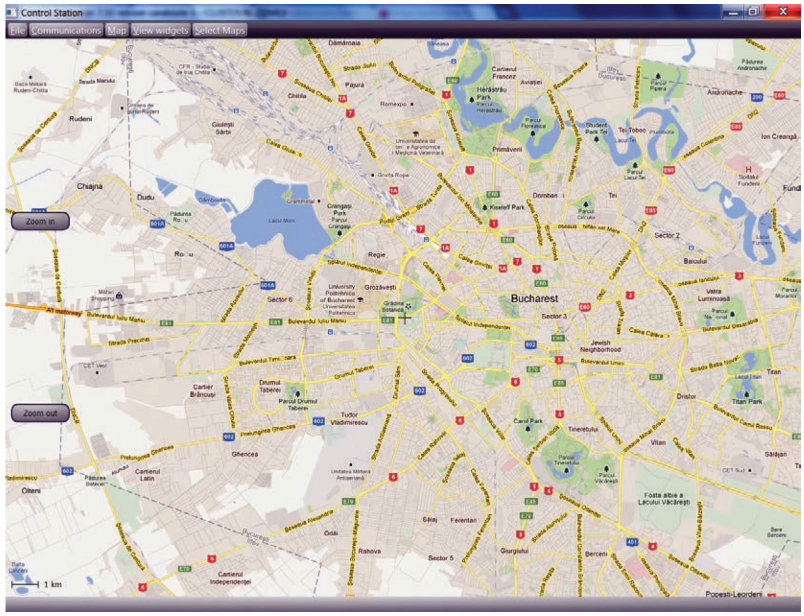


FIGURE 4. Global Map Control displaying the current GPS position.

control station and the robot. Between the two, communication is achieved by using a radio transceiver. Both the robot and control unit software were developed in Microchip's MPLAB32 IDE. Since the MCU has to serve several tasks at once — some of them being computational intensive and others having real time requirements — a Real Time Operating System (RTOS) was chosen to provide multitasking and also because it has the ability to meet a deadline. FreeRTOS (**www.freertos.org**) is a free real time operating system written in C that is designed for embedded devices and is able to be ported to several microcontroller architectures. We chose FreeRTOS because it is small and features a preemptive kernel that is highly configurable. That being said, we encourage you to look it up.

Hardware Abstraction Layer

A Hardware Abstraction Layer (HAL) was created on top of FreeRTOS and Microchip's PIC32MX Peripheral library. This abstraction layer hides the complexity of the hardware, and makes code easier to understand and less error prone. The HAL covers all the peripherals (MCU's UART, AT86RF212 transceiver — the chip used on PmodRF1, etc.), most of them being interrupt driven which results in non-blocking method calls.

Control Unit

The control unit is a bridge between the computer and the robot. It offers two choices of connectivity: Ethernet and serial. Uptime information and battery voltage are shown on the mounted LCD display.

To achieve its functionality, the control unit uses the following hardware:

- Digilent Cerebot32MX4
- Digilent PmodRF1
- Digilent PmodNIC

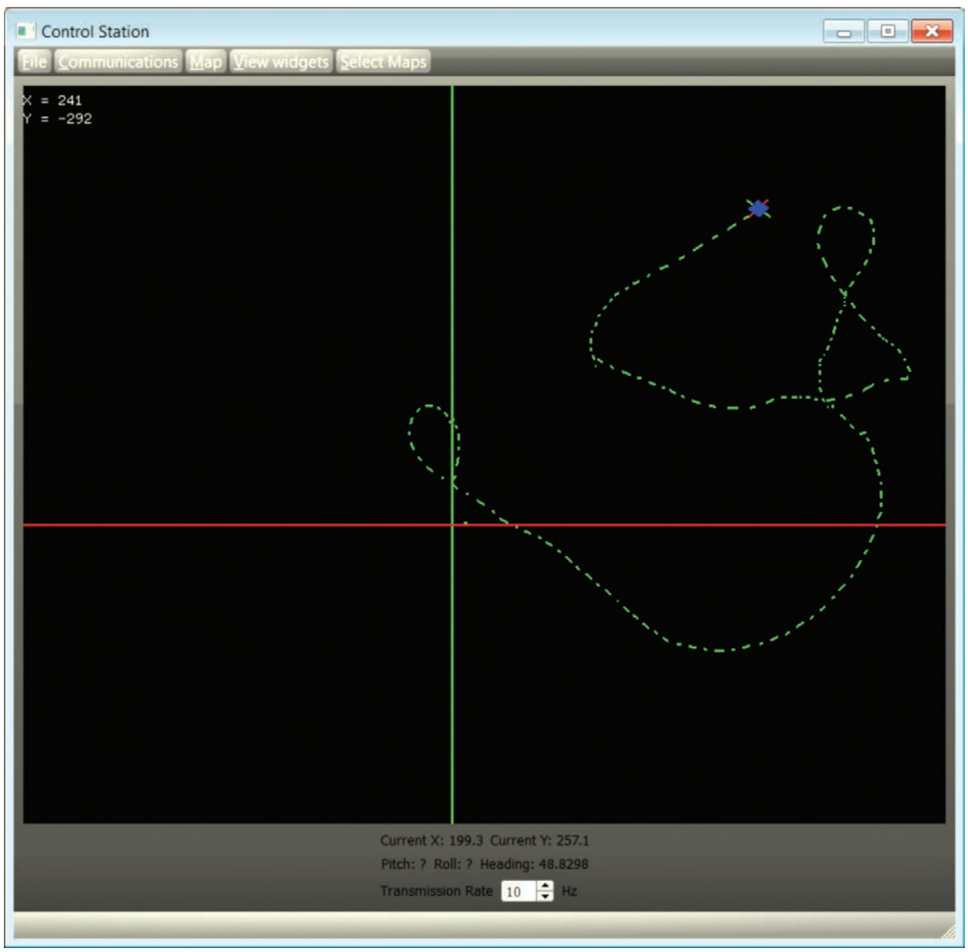


FIGURE 5. Local map widget displaying the robot's path.

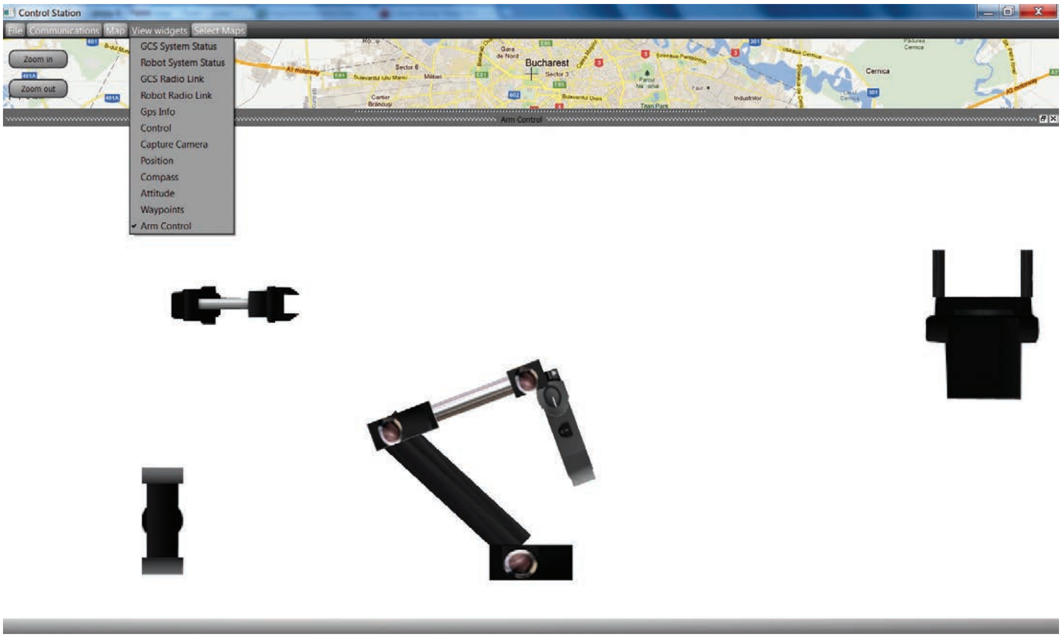


FIGURE 6. ArmControl widget displaying the current real position of the arm.

- Digilent PmodRS232
- NetMedia 4x16 Serial LCD (or Digilent PmodCLS)
- Digilent 5V VRM (Voltage Regulator Module)

Control Station Application

The robot's operator controls the system through a GUI application which offers the possibility to set up and visualize all the features offered by the rover. The application was designed with the aid of the C++ programming language and the Qt UI Framework. We chose Qt (**http://qt.nokia.com**) because it is free, simple, multi-platform, and very well documented.

The GUI application presents a user-friendly interface which is composed of a main menu, a central panel, and a series of floating widgets which offer the possibility to configure and visualize the results of the robot's features.

One of the most exciting widgets is the Local Map widget which allows the operator (in the absence of a GPS fix) to set waypoints in a local map relative to the robot's starting point, upload them, and visualize the robot's path.

ArmControl Widget

The ArmControl widget represents the software interface between the operator and the robotic arm. It consists of an OpenGL (**www.opengl.org**) based animation of the robotic arm and provides a means to control the arm using the wheel mouse. Its functionality is achieved by employing an inverse kinematics algorithm which determines the arm's position. Inverse kinematics is the process of determining the parameters of a jointed flexible object (a kinematics chain) in order to achieve a desired position.

Hardware Design

The following hardware parts were used to design the robot:

- Digilent Cerebot32MX4
- Digilent PmodRF1
- Digilent PmodTMP
- 4 x Digilent PmodHB3
- 4 x DC Gearhead motors with fitted encoders
- IMU module (three-axis accelerometer, three-axis gyroscope, magnetometer, barometer)
- Two servomotors for the pan/tilt mechanism
- A mechanical arm – Lynxmotion AL5C robotic arm
- Two Devantech SRF04 sonar range finders
- A4WD1 Lynxmotion robot chassis

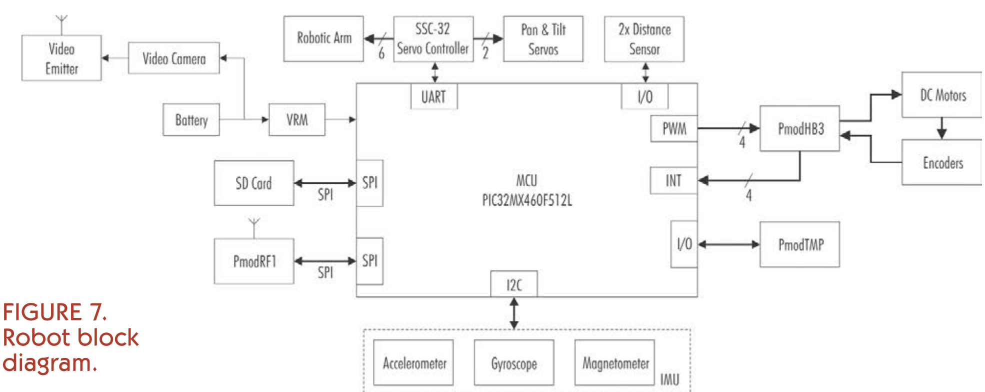


FIGURE 7. Robot block diagram.

A classic four wheel drive differential drive system was chosen. The aluminum chassis houses four DC gearhead

motors with built-in quadrature encoder, featuring 624 pulses per output shaft revolution. Differential wheeled robots present some advantages and disadvantages. Steering is achieved by varying the relative rate of rotation of the wheels and since it does not require an additional component for steering, it simplifies the navigation.

Since one of the main goals was to make the robot capable of achieving certain tasks without the operator's assistance, we implemented a navigation module.

The navigation software module fuses the information coming from the wheel encoders and the AHRS module to calculate the current position in a right-handed Cartesian coordinate system, centered in the robot's initial position with one millimeter used as scale.

The formulas used to calculate the current position were taken from G.W. Lucas's paper, "A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators" (**http://rossum.sourceforge.net/papers/DiffSteer**).

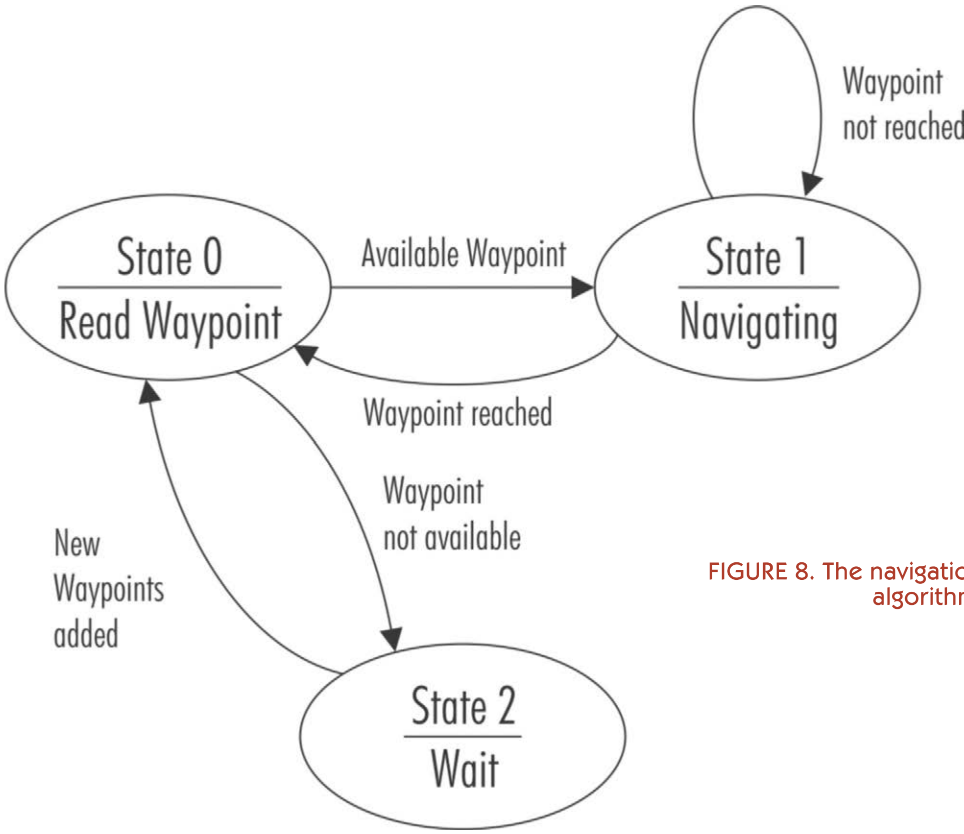


FIGURE 8. The navigation algorithm.

This software module has a finite state machine style implementation with three states.

PID Controller

Early in the development phase, we noticed that although the motors were of the same type they presented mechanical differences. Because of this and due to the fact that stable speed not influenced by the environment, load, or battery charge level was required, a PID control loop module was implemented.

A PID controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs. The feedback is provided by the quadrature encoders fitted on each DC motor. With every wheel revolution, the encoder generates 624 pulses, and every pulse generates an interrupt in the MCU which counts every rising edge of these pulses.

Four PID control loops were created; one for each motor. The PID constants were obtained using Matlab simulation. In order to calculate these constants, a series of bench experiments were conducted.

Conclusion

In this article, we've just scratched the surface of what is needed to build a remote controlled and autonomous robot. We encourage you to visit Digilent's website or the article link and download the source code for both the robot and the Control Station GUI. **SV**

The Smart Shopping Cart

by Chuan Zhang and Hairong Yan
Embedded Software and System Institute School of Software Engineering
Beijing University of Technology
Beijing, China

www.servomagazine.com/index.php?/magazine/article/february2012_Zhang
Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

FIGURE 1. The smart shopping cart.

This entry in the 2011 Digilent Design Contest is for a smart shopping cart system based on multi-sensor, radio frequency identification (RFID), visual identification, and pulse width modulation (PWM) control. It uses Digilent's Nexys2 FPGA board. Infrared sensors and ultrasonic sensors provide distance information. A joystick and wireless camera in the handle of the cart enable users to either drive or be followed by the cart. By using the RFID tags as an electronic key, the cart can record the user's identification information and accomplish self-checkout. The smart shopping cart system achieves low power consumption, low cost, and prolonged operating time.

Why This Project?

A lot of people go to supermarket and have to push a big bulky cart around. So we thought, "Why can't we develop something more intelligent, more interesting, and more enjoyable to use?"

Hardware Description And System Theory Overview

We began with a top-down design approach. For each functional module, we assigned a corresponding IP core. All data collected from the IP core was then processed in a Spartan 3E FPGA core. A modular and VHDL code design enhances the whole system's real time and reliability

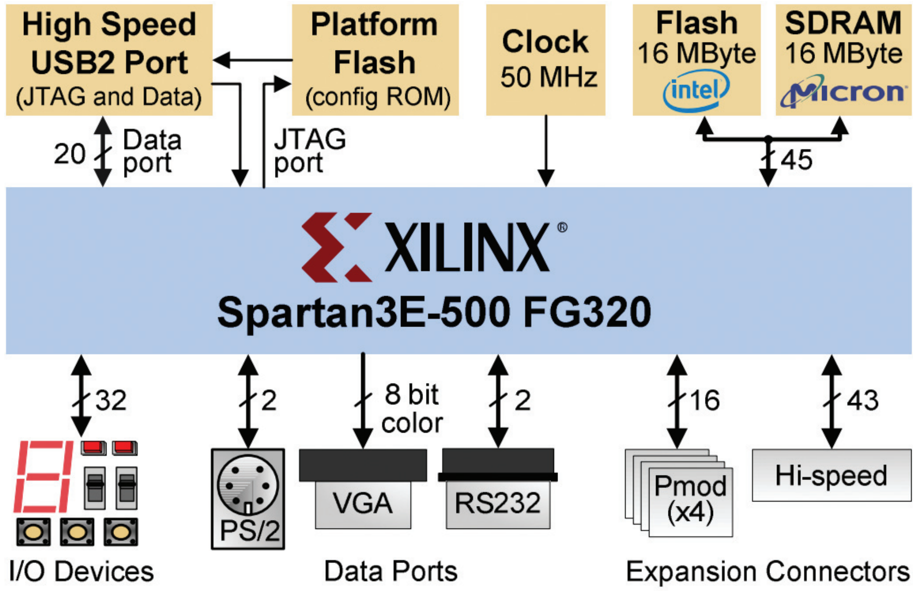


FIGURE 2. Block diagram of the FPGA development board.

features, and reduces power consumption.

Hardware Description

The central processing unit of the smart system is the Digilent Nexys2 FPGA development board. The development board — based on a Xilinx Spartan 3E FPGA — includes integrated USB2.0 ports, a large number of I/O acquisition ports, a data port, and expansion connectors.

We used FPGA as the main chip and transformed an ordinary supermarket cart. This transformation can be divided into two basic categories: electronic parts and mechanical parts.

Electronic parts include infrared sensors to detect objects, ultrasonic sensors to detect the distance between the cart and people, a wireless video camera to capture images for the tracking algorithm, supermarket electronic scanning to get information on products, an LCD to display the price, and RFID and wireless communication to exchange data with the server. Mechanical parts include the cart chassis assembly and motor drive.

System Theory

This system exploits the FPGA's programmability, as well as hardware implementation advantages. We use parallel data acquisition instead of serial data collection. System functionality is achieved with six modules: the top-level module, PWM motor control module, occurrence and acceptance of ultrasonic info module, LCD display module, supermarket electronic scanning module, and wireless communication control module. The FPGA system design worked from the top down, using a component-oriented programming method. By using this method, functionality of each module can be independently written, debugged, and optimized.

The top-level module not only sends and receives control signals from the lower function modules, but also controls the motors in different states (e.g., manual driving state and automatic tracking state) to meet the demands of the user. There are always different control signals coming from different I/O interfaces in different states. So, by aggregating the various control signals of functional modules to the top module, the state machine can use these signals to control the switching between the different states. The top module also controls all the actuators (e.g., motors, voice playback switch, etc.) in different states.

For different functional modules, the system makes efficient use of the FPGA's interface features. For example, the Zigbee wireless communication module is connected with FPGA through the RS-232 interface. This is an easy

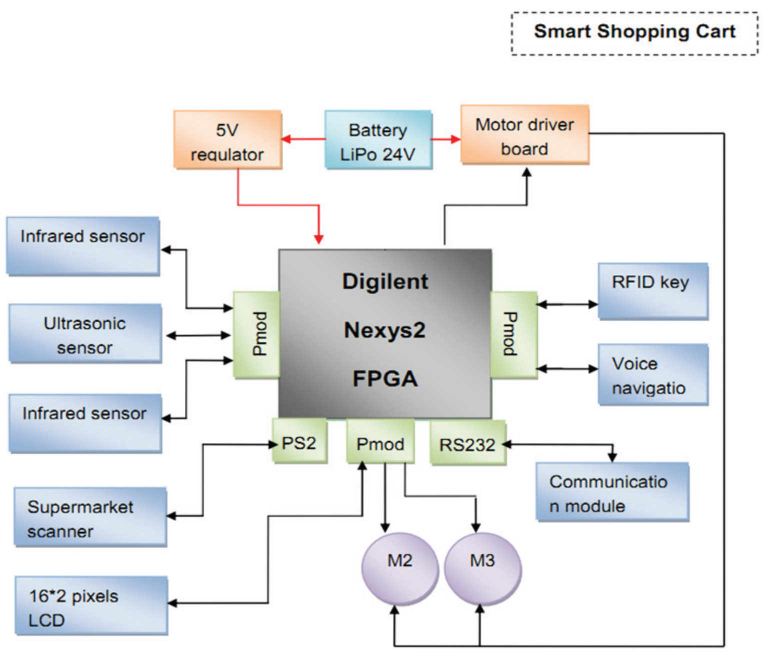


FIGURE 3. The system function module interface diagram.

FIGURE 4. Manual driving demonstration.

way for the RS-232 IP core to be supported by the Xilinx Spartan 3E; we only need to call the IP core and change some serial port parameters to achieve any related functions, so operation is very simple. The supermarket scanner module uses a PS/2 keyboard interface on the FPGA. This is because the scanner output code signal is the same output signal as the number keys on the keyboard. The other module's interfaces to the system use the Pmod interfaces on the FPGA. Pmod interfaces are reserved for additional sets of general-purpose input/output interfaces; these are flexible and can link directly to the FPGA chip pins.

Cart Functions

Through the development of the hardware and software, the smart shopping cart achieves manual cart driving, automatic following, an RFID electronic key, a voice shopping guide, and self-checkout. Following are brief descriptions about these features.

Manual Cart Driving

A game joystick is installed on the handle of cart so users can control movement style in all directions, including forward, backward, left, and right. The cart will automatically stop when sensors detect obstacles on any side. The cart's running speed can be adjusted to ensure safety of operation. When customers feel tired during shopping, they can simply stand on the supermarket cart and switch it to driving mode.

Automatic Following

The smart shopping cart provides a new shopping experience: just release your hands and focus your attention on the products you need to buy; your supermarket cart will follow behind you. In case the smart cart is faced with different environments, it has been set up with infrared sensors and a camera for its following mode. The infrared following mode is simple to implement. The tracking effect is good and isn't easily impacted by interference in the environment. One issue is that the infrared mode cannot distinguish between different people; the cart will follow any moving objects that appears in front of the cart. This error may occur unexpectedly, and is difficult to autocorrect.

One way to avoid this problem is to join the camera mode with the infrared sensor mode. When using the image tracking algorithm and the wireless camera module, the smart cart will follow the initial target image during the following process. The algorithm distinguishes the target image to the rest of the environment, based on relevant physical characteristics. By using algorithms, the smart cart can automatically find the target image in the environment, achieve a track positioning, and adjust itself to the correct location. With a combination of the two tracking modes, the cart can achieve automatic tracking and automatic correction.

Voice Shopping Guide

To help make the supermarket

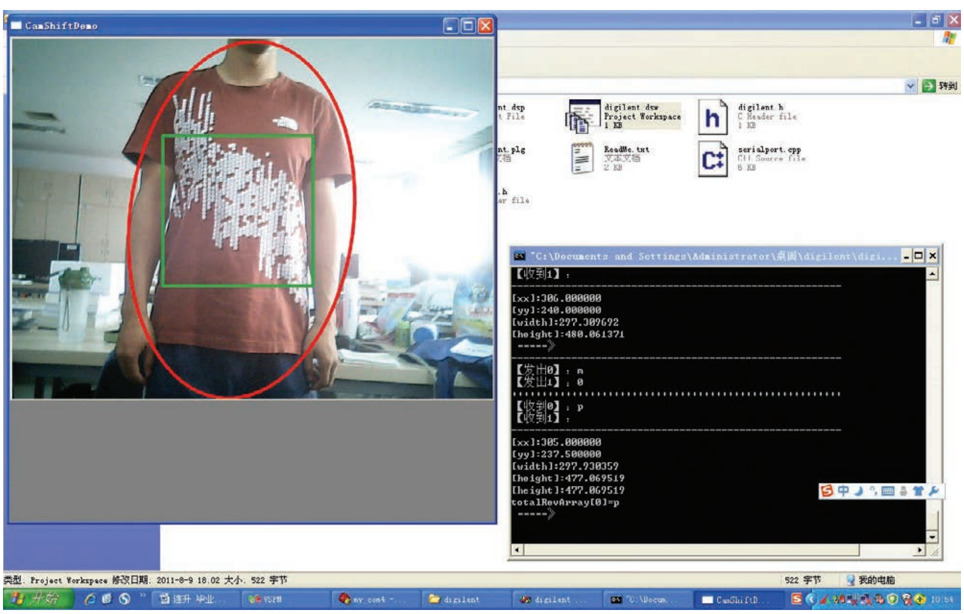


FIGURE 5. The automatic following control interface.

FIGURE 6. The self-checkout feature.

shopping process simple, we made use of an existing voice module. Since the FPGA is connected to the voice control unit by the Pmod interface, users can play or stop playback themselves. The voice module provides information on discount or promotional products for the particular store. In the future, all the information can be updated or modified via the wireless module. The voice module will be able to prompt users regarding the total price of the items they're purchasing.

Self-checkout

The self-checkout feature combines with the RFID tag technology, the supermarket scanner to read the product identification information, the LCD digital display, and the wireless communications functions. RFID is the basis for achieving self-checkout; RFID's also used as an electronic key, so it can start or lock the cart. In the future, supermarkets could have customers set up accounts with relevant personal information. Then, customers can choose to associate this information with with a credit card or prepaid funds they can access in a unique identity account.

Users would simply scan codes of items they want before the supermarket scanner. Information about the items, like the name and price will display on the LCD screen. Users would then choose to confirm or cancel the data before it's sent to the server by the wireless communication module. If they confirm the data, the system will automatically record and store the appropriate information to the FPGA memory cell through the wireless transceiver module (Zigbee or others). The server can then record the final purchase list and deduct the amount from the RFID related account.

Sources

Smart shopping cart video link:
http://v.youku.com/v_show/id_XMzAxNTAzNTk2.html

FPGA development board related references:
www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,789&Prod=NEXYS2

Conclusion

A smart shopping cart can achieve self-checkout functions and also pioneer a new shopping model of having a cart automatically follow a user. In order to improve the practicality and security of an intelligent cart, the system must integrate the use of RFID radio frequency tags, multi-sensor obstacle detection technologies, and FPGA as its control unit. This system makes full use of the abundant FPGA hardware resources, such as re-programming, IP multiplexing, and rich I/O interfaces, and also reserves enough hardware resources for future expansion. SV

Giving Your Robot Ears

Ways to command your robot using sound, music, and voice

by Gordon McComb

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

In space, no one can hear you scream . . . at your robot.

Happily, there is no such problem here on earth where you can use noise, sound, even voice to communicate your desires to your robotic plaything. All it takes is the right electronics — much of it already comes in convenient and affordable plug-in modules. Add an Arduino as the control brain and your robot now has ears to listen to the world around it.

In this article, you'll discover several methods for using sound to control your Arduino-based robot. You'll learn how to make your robot respond to loud noise; how to tell the difference between a soft footstep and a pile of dishes crashing to the floor; ways to communicate using a whistle or musical instrument; and even how to shout out commands for your robot soldier to follow. There's lots to cover, so let's begin.

Building on the ArdBot Base

As a brief note, I've used the ArdBot base (detailed in the November '10 through May '11 issues of *SERVO*) as the demonstrator platform for all the projects presented in this article. The ArdBot base provides a simple and low cost method of experimenting with Arduino robotics. The base is easy to construct (or you can get the parts premade at Budget Robotics; see the **Sources** box) and uses two continuous rotation R/C servos for motors.

Figure 1 shows a completed ArdBot, ready for all the projects presented here. It uses two decks, making expansion easy. The top deck has room for an Arduino Uno or compatible microcontroller, solderless breadboard, and several sensors and accessories. Power comes from two sets of batteries: a nine volt cell for the Arduino and a pack of four AAs for the servo motors. The batteries are mounted on the bottom deck.

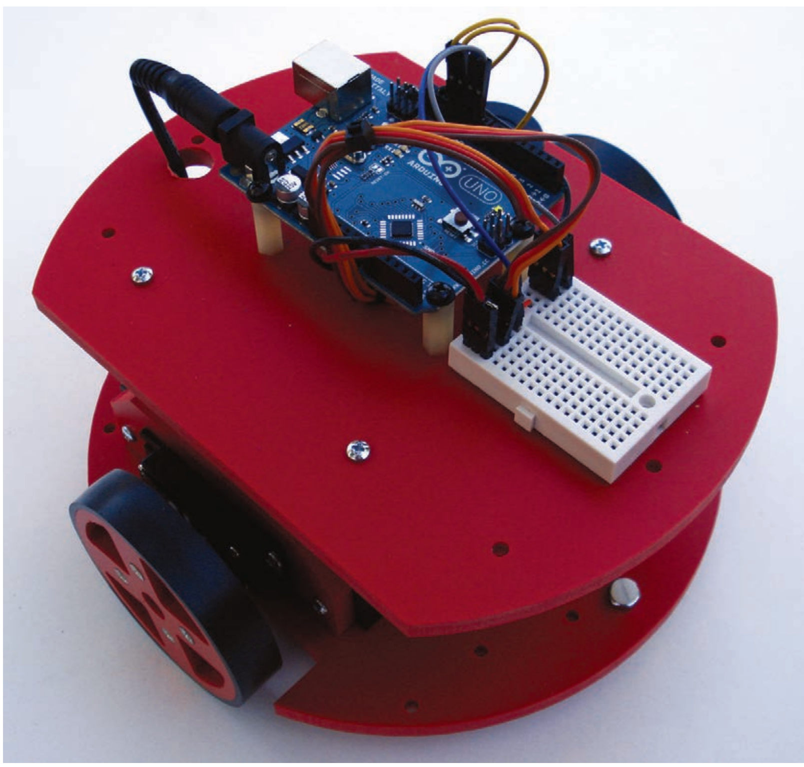


FIGURE 1. ArdBot servo motorized robot platform, ready for all the projects in this article. Batteries for the Arduino and motors go underneath.

Responding to Loud and Abrupt Sounds

Remember the "clap on, clap off" light switch control from a few years back? You know, clap once and the lamp beside your easy chair turns on; clap again and the lamp goes out. You can do the same thing with your ArdBot by using a noise activation sensor. These compact modules incorporate a microphone, amplifier, and all the necessary interface electronics. A small trimmer

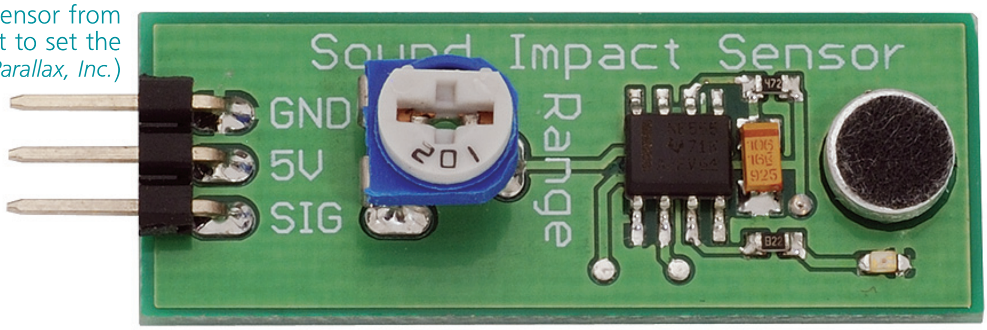


FIGURE 2. The Sound Impact Sensor from Parallax. Adjust the trimmer pot to set the triggering level. (*Photo courtesy Parallax, Inc.*)

potentiometer lets you set the sound volume that triggers the sensor. **Figure 2** shows the Parallax sound impact sensor which is both low cost and easy to connect to the Arduino or other microcontroller.

Though sound is analog, this type of noise activation sensor provides a digital off/on output signal. The signal is normally off (LOW) when the sound is below the threshold level you've set with the trimmer pot. Sound above the threshold causes the output to turn on (go HIGH).

Figure 3 shows how to wire the sound impact sensor to digital pin D2 on the Arduino. This pin is specifically selected because it is one of two Arduino I/O pins that support external interrupts; a change on this pin can trigger a function in the sketch. By using an interrupt, the Arduino doesn't need to constantly poll the I/O pin to determine the state of the sensor.

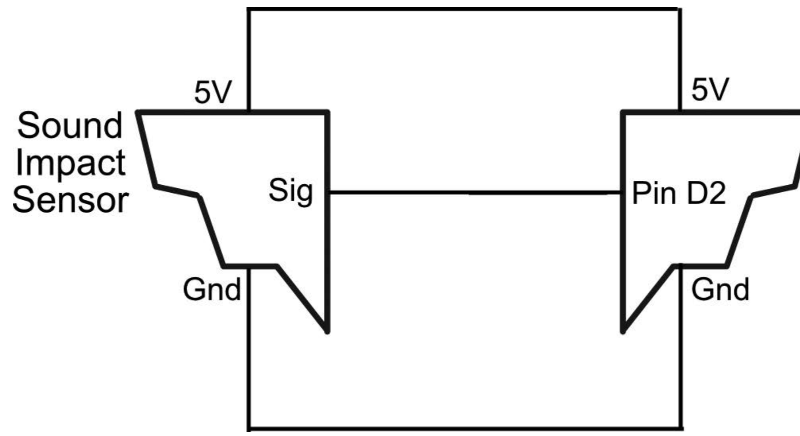


FIGURE 3. Simple wiring connection between the Sound Impact Sensor and the Arduino.

Polling requires the sketch to constantly monitor the status of the input pin. If a sound event is very short and occurs between polls, the Arduino may miss it. The more tasks handled by the Arduino in its *loop()* function, the more chance sound events will go unnoticed.

Interrupts are monitored in the background. The interrupt is triggered only if and when a change occurs to the input pin. As sound events can be fleeting, using an interrupt to process them is the best method to ensure that each one is captured.

Refer to **Listing 1** for a demonstration of using the Parallax sound impact sensor with an Arduino interrupt. There are two important pieces that make the sketch work: First is the *attachInterrupt* statement, found in the *setup()* function. This statement sets up the interrupt, specifying the interrupt number (interrupt 0 is on pin D2, interrupt 1 is on pin D3); the name of the function that is executed when an interrupt occurs; and the type of signal change that causes the interrupt to trigger. The second important piece is the interrupt function itself — called an *interrupt handler* or *interrupt service routine (ISR)* in programming parlance. In the sketch, the function is named *sounder*.

The **Listing 1** sketch demonstrates how each event that triggers the sound impact sensor toggles the Arduino's LED on or off. Set the trimmer pot on the sensor to adjust its sensitivity. Turning it clockwise reduces sensitivity; turning it counterclockwise increases sensitivity. Use the built-in LED on the sensor to help you adjust the trigger level.

Here's a tip: Collisions with objects cause noise. You can use a sound impact sensor to detect if your ArdBot has crashed into something. Increase sensitivity to hard contact by mechanically fastening the sensor to the base of the robot. Adjust the trimmer pot on the module to capture the sound of running into an object. Avoid setting the control so low that it will trigger on the robot's own motors.

LISTING 1 – SoundImpactDemo

```
const int soundPin = 2;
const int ledPin = 13;
volatile int state = LOW;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  attachInterrupt(0, sounder, RISING);
}

void loop() {
  digitalWrite(ledPin, state);
}

void sounder() {
  state = !state;
  delayMicroseconds(1000);
}
```

Extending Sound Triggering to Control Motors

Now take a look at **Listing 2**. It takes the basic concept

LISTING 2 – SoundImpactMotor

```
#include <Servo.h>

Servo servoLeft;           // Define left servo
Servo servoRight;          // Define right servo

const int soundPin = 2;
const int ledPin = 13;
volatile int state = LOW;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  attachInterrupt(0, sounder, RISING);
  servoLeft.attach(10);    // Left servo on pin D9
  servoRight.attach(11);   // Right servo on pin D10
}

void loop() {       // Loop through motion tests
  if (state) {
    botForward();   // Move forward
    delay(2000);    // Wait 2000 milliseconds (2 seconds)
    botReverse();   // Move backward
    delay(2000);
  } else {
    botStop();
  }
}

// Motion routines for forward, reverse, stop
void botForward() {
  servoLeft.write(180);
  servoRight.write(0);
}
void botReverse() {
  servoLeft.write(0);
  servoRight.write(180);
}
void botStop() {
  servoLeft.write(90);
  servoRight.write(90);
}

void sounder() {
  state = !state;
  digitalWrite(ledPin, state);
  delayMicroseconds(1000);
}
```

Be Sure to Update Your Arduino IDE

All of the program listings in this article were developed and tested using the Arduino 1.0 IDE software. If you're still using a pre-release version, be sure to visit the arduino.cc website and download the latest.

Among some of the more important changes: Beginning in version 1.0 sketch names now end in .ino, rather than .pde. Some of the Arduino commands — like the *Serial.print* method — now work a bit differently. Remember that you can keep older versions of the Arduino IDE on your computer, in case you need to go back and compile a sketch using an earlier edition.

of reacting to events from the sound impact sensor to operate the servo motors of the ArdBot. This sketch adds two *Servo* objects for the left and right servo motors, and a small assortment of functions for controlling the direction of the servos. The robot is started and stopped at each triggered sound event. The Arduino's LED (hard-wired to pin D13) shows the running status of the motors.

The sketch in **Listing 2** is kept simple to conserve page space. As written, the motors are turned off only after going through a sequence that involves several time delays. That means the motors may keep turning for a period of time after a sound event has occurred. There are a number of ways you can revise the sketch if this behavior isn't what you want. Here are two:

Detach and re-attach the servos within the interrupt handler. It's a good idea to keep the code inside an interrupt function as simple as possible, but adding the following shouldn't cause too much trouble:

```
if (state) {
  servoLeft.attach(10);
  servoRight.attach(11);
} else {
  servoLeft.detach();
  servoRight.detach();
}
```

Don't use delays to time the servo motors. Processing is effectively halted each time you use the *delay* statement. Delays *block* code from executing until the waiting period has expired. Actually, this is true only for programming outside of an interrupt. Code in an interrupt routine is executed without having to wait for the delay to expire. With this in mind, you can use an interrupt-based timer library — such as *MsTimer2* — to control the operation of the robot's motors. The motors are turned on and off based on timed events which do not block execution of code. (See the Libraries page on the *arduino.cc* website for details on this, and several other timer libraries you can use.)

Responding to Varying Sound Levels

A sound impact sensor is a quick and easy way to add ears to your bot. However, a drawback is that the sensitivity level is set manually. To respond to the instantaneous changes in sound level, use an amplified microphone coupled to one of the Arduino's analog inputs.

Personally, I detest building sound amplification circuits as they tend to be quirky and bulky. Though a bit more expensive, I prefer a store-bought microphone amplifier, like the SparkFun BOB-09964 (shown in **Figure 4**). The compact module — which is the size of a fingernail — is self-contained with an electret microphone element and amplifier circuit. It has just three connection points: ground, +V power (3.3V or 5V), and audio output.

The SparkFun microphone board is intended to be interfaced to a wide variety of electronics. While attaching to an Arduino is one of its intended applications, the connection requires a few additional components (shown in **Figure 5**). The capacitor and 1N34A diode — the latter a *germanium* type — form an unbiased DC clamp. As is typical of an audio amplifier, the output of the breakout board floats at a point midway between ground and +V power. The clamp restores the voltage in reference to ground (0V).

With the components shown and when powered by 3.3 volts (which I recommend), you can expect converted analog-to-digital values ranging from about 0 to 750 or 800 — the louder the sound, the higher the numeric value.

Listing 3 shows a working sketch that polls analog pin A0, and if it exceeds a preset threshold, displays the value in the serial monitor window. The threshold is purely for demonstration, so that only the louder sound events trigger a reading in the serial monitor (reduces clutter). Adjust the threshold variable to experiment. In a full sketch, you'd no doubt be interested in reacting to sounds of varying loudness; perhaps with an *if* or *case* statement to perform different actions, depending on the sound level.

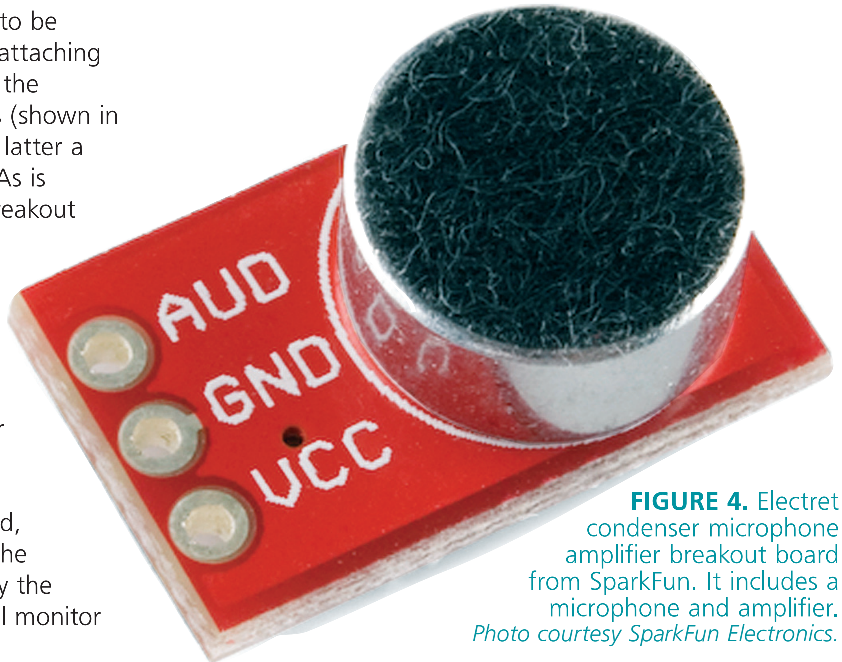


FIGURE 4. Electret condenser microphone amplifier breakout board from SparkFun. It includes a microphone and amplifier. *Photo courtesy SparkFun Electronics.*

Reacting to Different Sound Frequencies

Loudness (volume) is just one quality of sound your robot can listen for. Another is the frequency of the sound. Picture a piano keyboard, where the notes produce different sound frequencies that your robot can hear and follow.

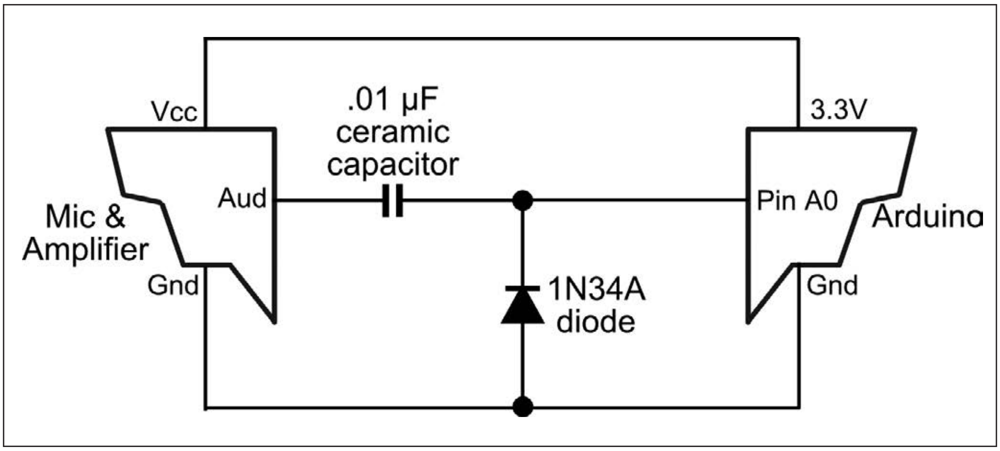


FIGURE 5. How to connect the microphone amplifier to the Arduino. Be sure the ground connections between the amp board and Arduino are shared.

You can combine a microphone and amplifier with a tone decoder circuit that differentiates the frequency you want from all other sound in the room. Fine idea in theory, but tone decoder circuits are notoriously troublesome. A better method is to use a specialty integrated circuit — an audio equalizer IC that provides the relative amplitude of a range of frequencies from 63 to 16,000 (16K) Hz.

The MSGEQ7 from Mixed Signal Integration, is an inexpensive eight-bit chip that requires only a few common external components, plus three connection wires to the Arduino. A sample schematic — derived from the MSGEQ7 datasheet — is provided in **Figure 6**. The MSGEQ7 is intended for use in building audio graphic equalizer displays like the kind on your stereo or home theater system. It has a single output pin that produces a voltage relative to the loudness of the sound.

The chip is sensitive to several bands with each band centered around the following frequencies: 63 Hz, 160 Hz, 400 Hz, 1 kHz, 2.5 kHz, 6.25 kHz, and 16 kHz. As a point

LISTING 3 – MicAmplifier

```
const int ledPin = 13;
const int soundPin = A0;
int threshold = 300;
int delayPeriod = 100;

boolean indicator = false;
int soundLevel = 0;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  soundLevel = analogRead(soundPin);
  if (soundLevel >= threshold) {
    indicator = !indicator;
    digitalWrite(ledPin, indicator);
    Serial.println(soundLevel, DEC);
    delay(delayPeriod);
  }
}
```

LISTING 4 – SoundEqualizeDemo

```
const int inputPin = A1;    // Analog 1; DC out (EQ pin 3)
const int strobePin = 3;    // D3; Strobe (EQ pin 4)
const int resetPin = 4;     // D4; Reset {EQ pin 7)

int audio[7];                // Array for EQ values

void setup() {
  Serial.begin(9600);
  pinMode(inputPin, INPUT);
  pinMode(strobePin, OUTPUT);
  pinMode(resetPin, OUTPUT);
  digitalWrite(resetPin, LOW);
  digitalWrite(strobePin, HIGH);
}

void loop() {
  digitalWrite(resetPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(resetPin, LOW);
  for (int i = 0; i < 7; i++)  {
    digitalWrite(strobePin, LOW);
    delayMicroseconds(40);
    audio[i] = analogRead(inputPin);
    digitalWrite(strobePin, HIGH);
    delayMicroseconds(40);
    Serial.print(audio[i], DEC);
    Serial.print("\t");
  }
   Serial.println();
   delay(250);
}
```

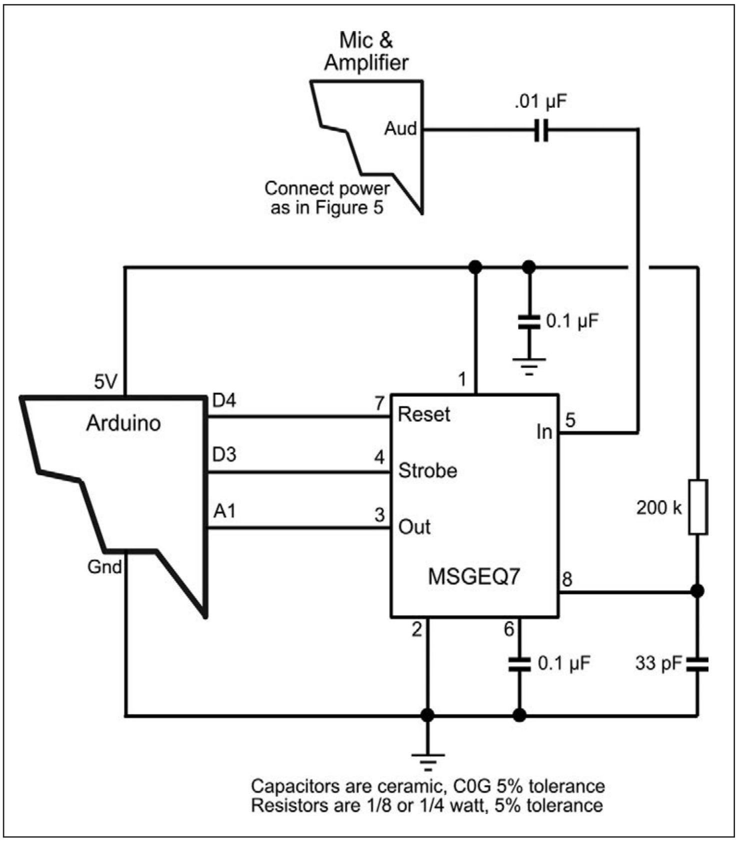


FIGURE 6. Wiring diagram for the MSGEQ7 graphic equalizer display filter chip from Mixed Signal Integration, and available from SparkFun. Resistor tolerances should be 5% or better; capacitor tolerances 10% or better.

of reference, the human voice spans a range of about 150 Hz to 1,250 Hz; the 88 keys on a piano range from 27 Hz to 4,187 Hz.

A sketch demonstrating how to use the MSGEQ7 IC is provided in **Listing 4**. The code displays (in the serial monitor window) the relative values of the seven frequency bands, from 0 to 1,023. Note that due to room and electrical noise, the low end of the scale commonly starts above 100. In operation, the sketch cycles through the seven bands one at a time, sampling the instantaneous voltage level from pin 3 (DC output) of the MSGEQ7 IC. Once all seven values have been collected, they're shown on the serial monitor window.

The EQ chip works best in quiet environments without conflicting sound sources. Ambient (natural) sounds — like the hum of a computer fan or even the chirping of birds outside — can influence the output. Even a simple sound source — like one key on a piano — produces many frequencies up and down the audio spectrum. This is due to such things as harmonics and sympathetic vibrations produced by nearby objects. To make better sense of the values returned by the MSGEQ7, you need to weight all seven bands and use that data to determine the center frequency of the sound being analyzed.

One very simple method of weighting the seven values is to determine which one is highest. Here's a function that finds the index with the highest value (remember, Arduino array indices start at 0):

```
int getMax(int* array, int arraySize){
  int maxIndex = 0;
  int maxVal = array[maxIndex];
  for (int i = 1; i <arraySize; i++){
    if (maxVal < array[i]) {
 maxVal = array[i];
 maxIndex = i;
    }
  }
  return maxIndex;
}

Call the function with:

int index = getMax(audio, 7);
```

As frequency analysis is not 100 percent reliable, use this method only for non-critical robot control applications. With practice, you'll get better at it. Experiment with different kinds of frequency generators to control your bot. With some patience, you might be able to reliably reproduce a couple of distinct whistles. Or, use a small plastic musical recorder as a kind of Pied Piper controller for your bot. How about a harmonica or Boatswain's whistle? See what works best.

Operating Your Robot by Voice Command

The ultimate in robot sound control is voice command: Say a word and your robot obeys. Voice recognition is common in desktop computers and even some smart phones, and with a separate module you can add the technology to your own robot. Of the several commercial products available, the EasyVR co-processor board from VeeAR is among the least expensive and easiest to use. It's available from a number of online sellers; see the **Sources** box for more details.

For my ArdBot, I used a Parallax Say It Module (**Figure 7**) which is an older generation VRBot, also by VeeAR. The EasyVR is an updated version of the VRBot/Say It Module, but overall functionality is similar.

Figure 8 shows a wiring diagram. There are just five pins to connect: power, ground, serial transmit and receive, and an optional LED indicator. The LED is under your software control. For the sketch in **Listing 5**, I'm using the LED to show when the module is ready for a new voice command. If a recognition or other error occurs, the LED flashes, then the module resets and awaits a new command.

The EasyVR/VRBot/Say It Module comes pre-programmed to respond to 26 speaker-independent words. (You can add your own custom speaker-dependent words, but that process is more complex and not covered here.) The command words are separated into four sets; the sketch in **Listing 5** uses the words in command set 2: Right, Left, Up, Down, Forward, and Reverse. The motor function of each command is obvious, except for Up and Down which

LISTING 5 - Servo_VoiceRecognitionMotor

```
#include <Servo.h>
Servo servoLeft;            // Define left servo
Servo servoRight;           // Define right servo

#include <SoftwareSerial.h>
#include "protocol.h"

byte wordset = TWO;          // Wordset 2
byte timeout = TWO;          // 2 second timeout

const int ledActive = 5;    // LED control
const int rxpin = 6;         // Receive pin (Rx)
const int txpin = 7;         // Transmit pin (Tx)

SoftwareSerial serial_vr(rxpin, txpin);
char c;

void setup() {
  Serial.begin(9600);
  serial_vr.begin(9600);  // Serial to VR
  servoLeft.attach(10);   // Set left servo to digital pin 10
  servoRight.attach(11);  // Set right servo to digital pin 11
  pinMode(ledActive, OUTPUT);
  delay(100);

  c = vrQuery(CMD_BREAK);  // Wake up VR
  c = vrQuery(ARG_ACK);
  serial_vr.write(CMD_LANGUAGE);
  c = vrQuery(ZERO);
  serial_vr.write(CMD_TIMEOUT);
  c = vrQuery(timeout);
}

void loop() {

  digitalWrite(ledActive, HIGH);
  Serial.println("Command: left, right, up, down, forward, backward");
  serial_vr.write(CMD_RECOG_SI);    // Start recognition
  c = vrQuery(wordset);              // Set wordset
  digitalWrite(ledActive, LOW);

  switch(c) {
    case STS_TIMEOUT:
      Serial.println("(TIMEOUT)");
      ledBlink();
      break;
    case STS_ERROR:
      Serial.println("(ERROR) ");
      ledBlink();
      c = vrQuery(ARG_ACK);
      c = vrQuery(ARG_ACK);
      break;
   case STS_SIMILAR:
     Serial.print("Successful recognition: ");
     c = vrQuery(ARG_ACK);
     vr_2(c);
     break;
   default:
     ledBlink();
     Serial.println("Not recognized");
     break;
  }
  delay(50);
}

void vr_2(byte c) {
  Serial.print(c, DEC);
  switch (c) {
   case WORDSET_2_LEFT:
     Serial.println(" (LEFT)");
     botLeft();
```

Continued

LISTING 5 – Servo_VoiceRecognitionMotor *continued*

```
      break;
    case WORDSET_2_RIGHT:
      Serial.println(" (RIGHT)");
      botRight();
      break;
    case WORDSET_2_UP:
      Serial.println(" (UP)");
      botForward();
      break;
    case WORDSET_2_DOWN:
      Serial.println(" (DOWN)");
      botStop();
      break;
    case WORDSET_2_FORWARD:
      Serial.println(" (FORWARD)");
      botForward();
      break;
    case WORDSET_2_BACKWARD:
      Serial.println(" (BACKWARD)");
      botReverse();
      break;
  }
}

char vrQuery(byte cmd) {
  serial_vr.write(cmd);
  while (serial_vr.available() == 0) { }
  return (serial_vr.read());
}

void ledBlink() {
  digitalWrite(ledActive, HIGH);
  delay(125);
  digitalWrite(ledActive, LOW);
  delay(125);
  digitalWrite(ledActive, HIGH);
  delay(125);
  digitalWrite(ledActive, LOW);
  delay(125);
}

// Motion routines for forward, reverse, turns, and stop
void botForward() {
  servoLeft.write(180);
  servoRight.write(0);
}
void botReverse() {
  servoLeft.write(0);
  servoRight.write(180);
}
void botRight() {
  servoLeft.write(180);
  servoRight.write(180);
}
void botLeft() {
  servoLeft.write(0);
  servoRight.write(0);
}
void botStop() {
  servoLeft.write(90);
  servoRight.write(90);
}
```

Sources

Budget Robotics
ArdBot Chassis: Precut Body Parts, Assembly Hardware
www.budgetrobotics.com

Parallax
Sound Impact Sensor (29132)
Say It Module (30080)
www.parallax.com

SparkFun Electronics
Breakout Board for Electret Microphone (BOB-09964)
Graphic Equalizer Display Filter — MSGEQ7 (COM-10468)
www.sparkfun.com

VeeAR
Easy VR Voice Recognition Module
(also sold through distributors, including SparkFun, Robosavvy, RobotShop, and others)
www.veear.eu

are used to turn the servos on and off, respectively.

Important! The sketch in **Listing 5** requires a separate constants file, *protocol.h*. The file is too lengthy to present here, but is included in the zip archive download that contains all the sketches in this article. Also included in the archive is a more robust demonstrator of the Say It Module. This demonstrator allows you to experiment with all four command sets. It also provides more verbose messages sent from the recognizer module.

The Say It Module uses an on-board microphone which is not ideal for robot control (the EasyVR comes with a microphone on the end of a short detachable cable). For best reliability, you should speak no further than 2-3 inches from the microphone, and the mic should be away from other sound sources. This pretty much precludes putting the module and microphone on the robot itself. Alternatives:

- Attach the microphone to a longer cable and hold the mic in your hands. The cable should be shielded and high quality, and not longer than about five feet. Even then, this approach may introduce too much noise in the microphone signal.
- Move the entire VR module off the robot and connect it to your bot via a wired link. Use twisted-pair wiring (the kind in Ethernet cables), and keep the wiring under 6-8 feet.
- Use a wireless link, such as XBee, to transmit serial commands between the VR and robot. This is a more expensive method, but the most reliable and flexible.

Word to the wise: Be sure to turn off the TV, radio, iPod, or any other sound-making source before experimenting with voice recognition. The system works best in a quiet room.

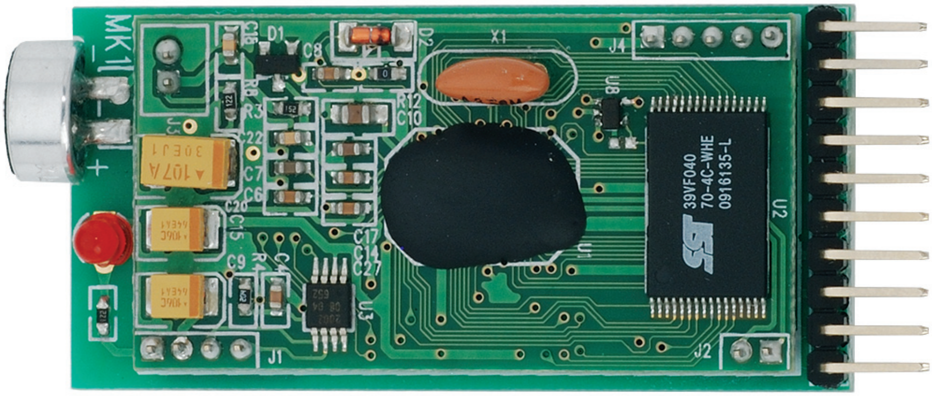


FIGURE 7. The Parallax Say It Module, a complete voice recognition system on one board. This module is similar to the VRBot and EasyVR products from VeeAR (all are made by the same company). *Photo courtesy Parallax, Inc.*

Sound In, Sound Out

Sound isn't a one-way street. In this article, we looked at ways to control a robot with sound, but you can also have your robot *produce* sound — and music and even voice effects. I've saved these topics for the next time, where you'll learn how to make your Arduino-based robot sound off. You'll discover ways to produce basic sound effects, play recorded or electronic music, and speak in a classic robotic voice. **SV**

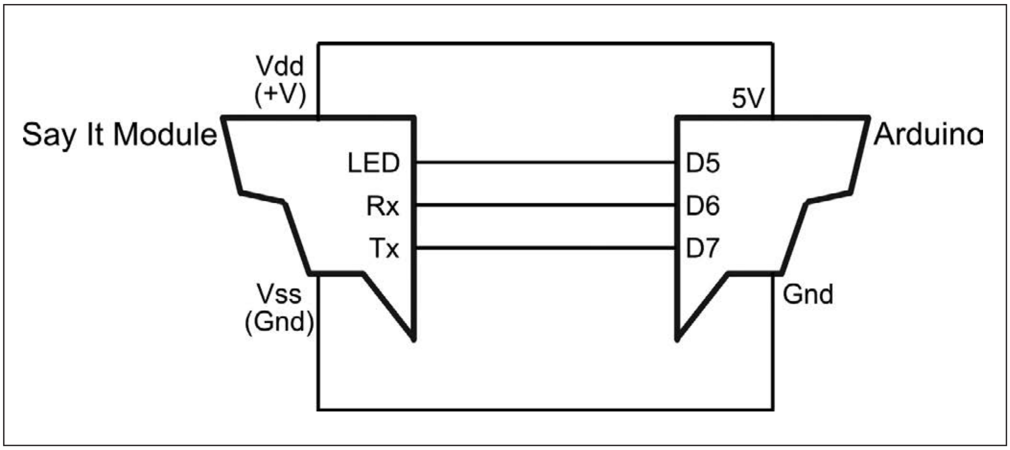


FIGURE 8. Wiring diagram for the Say It Module to the Arduino. The LED connection is optional, but recommended.

Product Review

Digilent NEXYS3 Spartan-6 FPGA Board

by David Ward

www.nutsvolts.com/index.php?/magazine/article/february_Ward

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

If you are interested in learning more about FPGAs (field programmable gate arrays) or implementing FPGA designs, Digilent (www.digilentinc.com) has several FPGA boards available. I recently became familiar with their Nexys3 Spartan-6 FPGA board. The Nexys3 board is built around the Xilinx XC6SLX16 Spartan-6 324-pin FPGA. The Spartan-6 family of FPGAs consists of 13 members; the XC6SLX16 is third up from the bottom of the group.

The Nexys3 board will allow you to explore many of the XC6SLX16's capabilities. The board has three types of non-volatile memory onboard which will allow you to have your configurations running on power-up.

It provides many industry standard connectors such as VGA, USB, Ethernet, JTAG, and a USB UART. It also features four 12-pin PMOD connectors which can be used to interface with Digilent Pmod devices or they can be used as 32 general-purpose I/O pins for your own circuit designs. A VHDC connector is provided which can be connected to proto-boards, cables, and other devices available from Digilent.

There are eight slide switches, eight LEDs, five pushbuttons, and four seven-segment LED displays on the board for experimentation. These items are a great way to get started using the Nexys3 board because they are easy to work with and do not require any external cables or connections.

Digilent provides a demonstration program (Nexys3_ISE_GPIO_UART.zip) from their Nexys3 web page that demonstrates the operation of these items, as well as sending out serial messages from the USB UART port. You will need to download a virtual COM port program from **www.ftdichip.com** in order to use this serial port feature. This is a free download which installs and operates without any problems.

The Spartan-6 FPGAs can implement Xilinx's MicroBlaze embedded processor designs. This allows you to configure and use the FPGA as a microprocessor.

All development software needed to program and interface with the Nexys3 board is free. You can use the latest Xilinx ISE web-pack software to compile designs using either a graphical schematic entry or VHDL (see **www.xilinx.com**).

To communicate with the board for testing purposes or to download configurations into the FPGA itself (or any of its non-volatile memories), Adept is used. Adept is available as a free download from **www.digilentinc.com**.

Documentation such as a reference manual and schematic diagrams are all available from the Nexys3 web page. The reference manual provides essential information about using all of the connections, memories, etc., on the board.

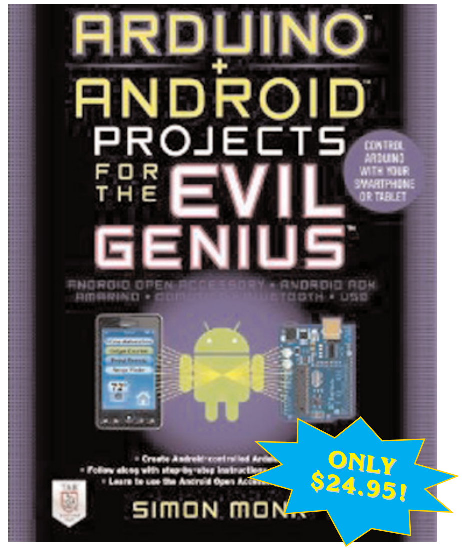
The Nexys3 board sells for approximately $199 or $119 for academic pricing. It's a great way to work with an FPGA at a reasonable price. **SV**

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS

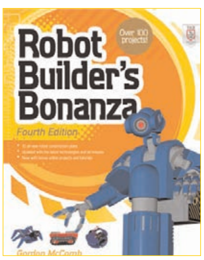
NEW RELEASE



ROBOTICS

Robot Builder's Bonanza, Fourth Edition

by Gordon McComb

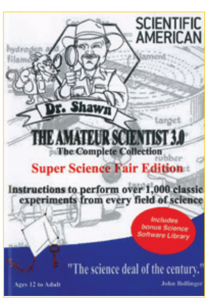


Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

$29.95*

The Amateur Scientist 4.0 The Complete Collection

by Bright Science, LLC

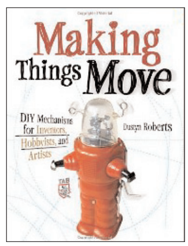


There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do your first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime!

$26.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts



In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

$29.95*

Build Your Own Humanoid Robots

by Karl Williams

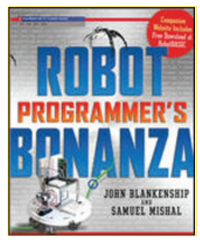
GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.

$24.95*

Robot Programmer's Bonanza

by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!

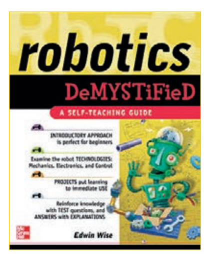
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

$29.95

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!



Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles

Published by T & L Publications, Inc.

$57 per bundle

Save $10 off the normal price!!

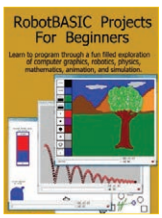
Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10 and 2011

Show your favorite geek some love this Valentine's Day!
Get them cool robotics stuff from my store!

Visit my online store @
www.servomagazine.com

RobotBASIC Projects For Beginners

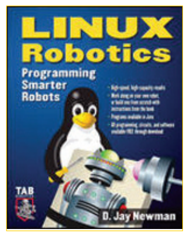
by John Blankenship, Samuel Mishal



If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text. **Price $14.95**

Linux Robotics

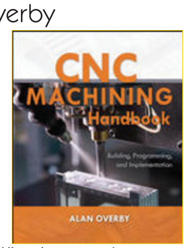
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need. **$34.95**

CNC Machining Handbook: Building, Programming, and Implementation

by Alan Overby



The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource. **$34.95**

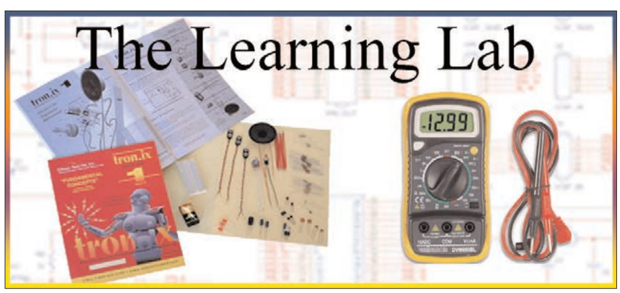
SPECIAL OFFERS

LEGO MINDSTORMS NXT Idea Book

by the Contributors to The NXT Step Blog

If you're serious about having fun with LEGO® robotics, you've come to the right place. The team behind The NXT STEP blog — the authoritative online source for MINDSTORMS® NXT information and advice — has packaged its considerable skills and experience in this book. Inside, you'll find some of the team's best ideas for creating cool and sophisticated models, including instructions for eight robots you can build yourself.

Reg $29.95 Sale Price $24.95



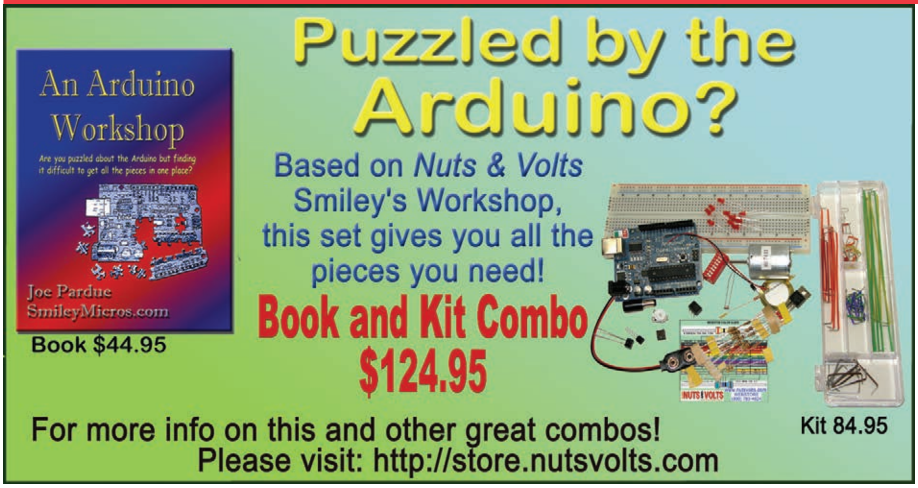
Technology Education Package for Everyone Starting in Electronics

This lab – from the good people at GSS Tech Ed – will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price $99.95 **Subscriber's Price $95.95**

Or order online www.servomagazine.com

SPECIAL OFFERS



Forbidden LEGO

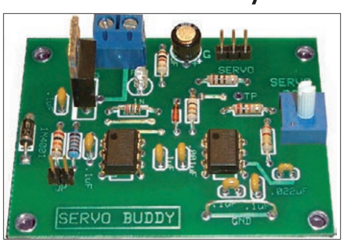
by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg $24.95 Sale Price $19.95

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

3D LED Cube Kit

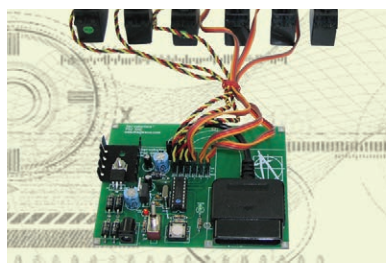
From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of Nuts & Volts Magazine.

This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue. Jig and plastic cases also available.

Subscriber's Price **$57.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.

For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**

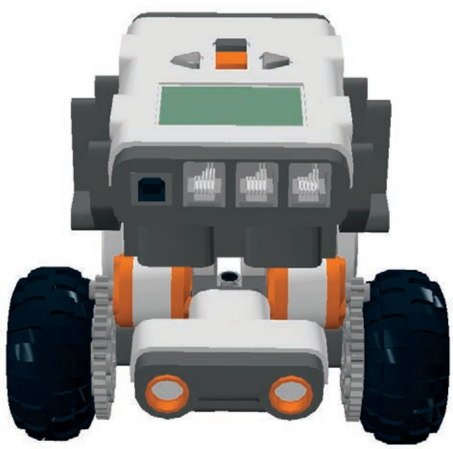
The NXT Big Thing #17

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Isotope — Part 3 of 3

By Greg Intermaggio

01001000011010010100001 (Binary for "Hi!") — Welcome back to The NXT Big Thing! If you're just joining us, you'll want to read back issues 15 and 16, where we designed, built, and began to program Isotope — an awesome all-terrain robot with LEGO MINDSTORMS NXT!



This month, we'll be adding the final functionality to Isotope's program that will allow his linear actuators to be remotely controlled, giving him the ability to traverse treacherous terrain. Let's break it dowwwwwn!

Converting the Bot Test Program

Refer to **Figures 1** and **2** here. There's a lot to be done here, but as long as we keep our heads about us, the additions to our Bot Test program should be a walk in the park.

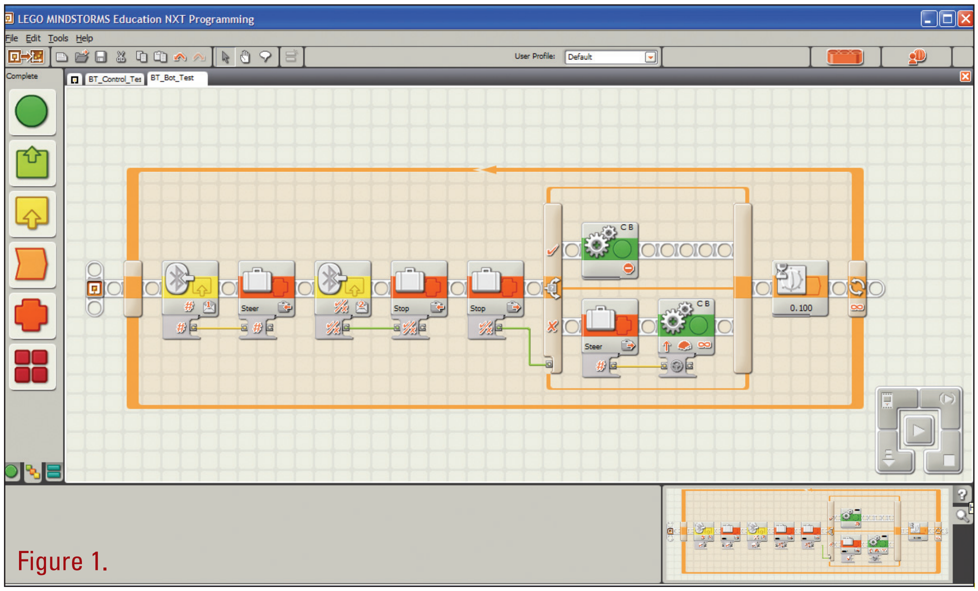


Figure 1.

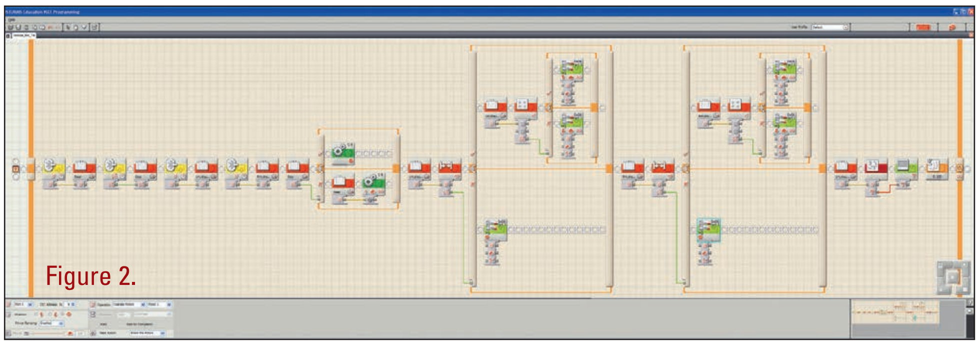


Figure 2.

- First, save your program as Isotope_Bot_Test.
- There are two new number variables: LHydraulics and RHydraulics. (Note: Our linear actuators are not actually hydraulic, but they look kind of like car hydraulics, hence the name!)
- Create a new Receive IR Message block to write to each variable. Pull LHydraulics from mailbox 3 and RHydraulics from mailbox 4.
- Read LHydraulics and use a Compare block to see if it's outside of the range of -5 to 5. Wire that block to a logic switch.
- If the LHydraulics variable is outside of that range, it means that the user wants the left actuator to move. So, in the true side of the switch read the LHydraulics variable; if it's less than 0, move the left actuator (motor port 1 on our NXTMMX) forward at full power for unlimited duration. Similarly, if LHydraulics is greater than 0, move the same motor backwards with an unlimited duration.
- On the other side of the switch, stop motor 1 on the NXTMMX. This will stop the actuator from moving (more on this later).
- Copy the switch we just created along with the read variable and compare blocks. Switch each instance of LHydraulics to RHydraulics, and each instance of Motor Port 1 to Motor Port 2.
- If you'd like to troubleshoot your program, you can read one of the variables. In my case, I read LHydraulics and displayed it on the NXT so I can verify that Isotope is receiving the variable correctly.

That should do it for the Isotope side of the program. Now, let's tackle the beast of the controller program. There's a lot to do here!

Converting the Control Test Program

Refer to **Figures 3** and **4**.

- First, save your program as Isotope_Control_Test.
- The first addition to our control test program is relatively easy. Add the following six number variables: BDiff, BTotal, CDiff, CTotal, BLast, and CLast.
- Read the BTotal variable and send it as an IR message to mailbox 3. Read the CTotal variable and send it as an IR message to mailbox 4.
- If you want to troubleshoot, you can view one

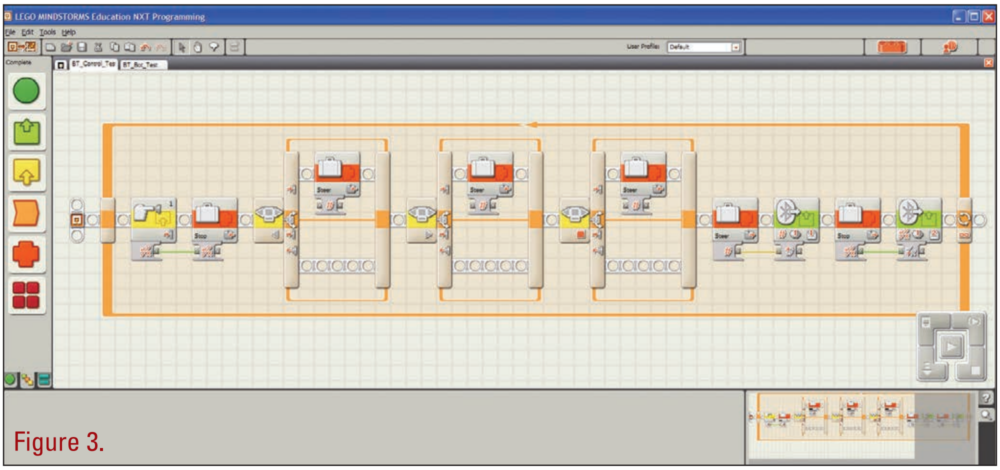


Figure 3.

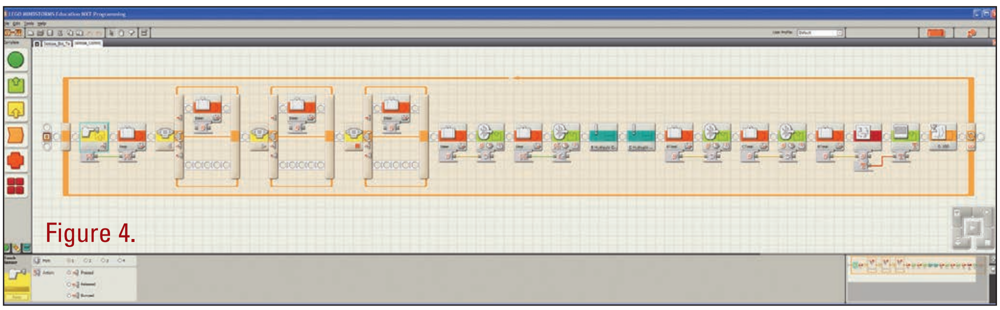


Figure 4.

subtract 3 from BTotal.

- Select the entire program and go to Edit > Make a New My Block, and finish the My Blockcreation process. I recommend naming this block B Hydraulic Control.
- Now, create the exact same program except use C for everything. Motor Port C, CLast, CDiff, CTotal, and name it C Hydraulic Control.

Final Touches

- Now, add the two My Blocks to your Isotope Control program as indicated in the image.
- Download the Isotope Control program to the remote control you built last month. Download the Isotope Bot program to Isotope itself.
- Connect the two NXTs via Bluetooth, and give the programs a run.

Isotope should have its normal forward/turn/stop obility using the NXT buttons on the controller, but now spinning the motors on the controller should cause Isotope's linear actuators to move in and out proportionally based on how much you spin them.

of your variables on the NXT screen. In this screenshot, I viewed BTotal to verify that it was returning the numbers I wanted, and it wasn't malfunctioning. More on this later!

The Tricky Part

Refer to **Figures 5** and **6**.

Now, we're going to need to create two My Blocks. These are custom blocks you can use in your program that are part of the main program, but are too big and need their own space to keep from making the program unmanageably large.

- Create a new program. Add the number variables: BLast, BDiff, and BTotal.
- Read variable BLast and use the Rotation Sensor block to read the motor on port B of your controller. Subtract BLast from the current motor reading and write the result to BDiff. Now, write the motor reading on port B to BLast.
- Add variables BTotal and BDiff together and write the result to BTotal. Then, read BTotal and run it through a Compare block to see if it's less than 0. Attach the result to a logic switch.
- In the true section of the logic switch, add 3 to BTotal. In the false section of the logic switch,

Undertstanding the Program

This is by a good margin the most complicated program we've done to date in The NXT Big Thing, so bear with me as I try to break down what happens.

Let's look at the control program. Specifically, let's look inside our B Hydraulic Control My Block. The first thing to understand is that we need to be able to compare the current state of the motor on port B to its state the last time the My Block was run (in this case, it runs every .10 seconds in our main program). By comparing its current state to its "BLast" state, we can find the difference between the two, and understand how far the motor has turned since we last recorded a change. This way, we know exactly how far the motor has moved in total which is important in telling Isotope how far to extend or contract its actuators.

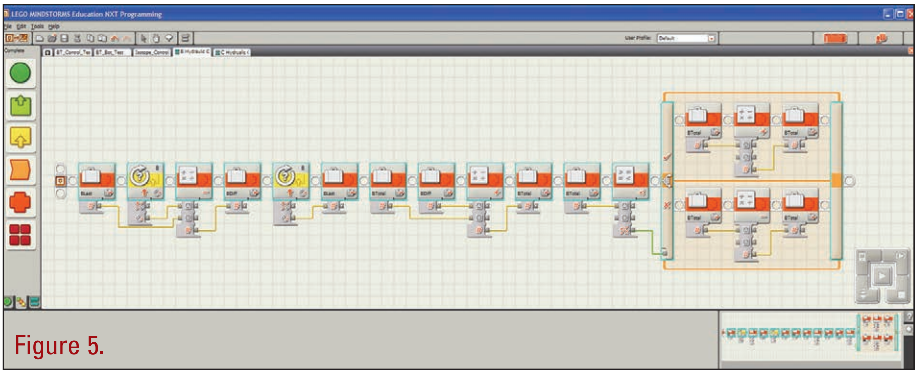


Figure 5.

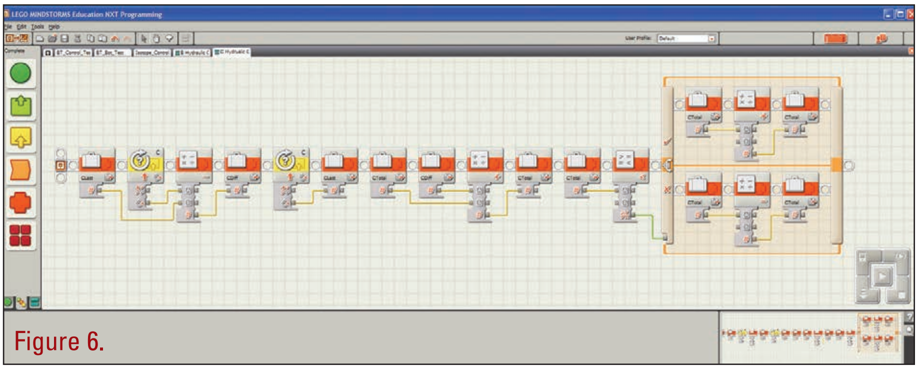


Figure 6.

We next add the difference or "BDiff" to a variable called "BTotal." This is an important step, as we use BTotal to determine the total movement of the motor. Eventually, if the motor has moved forward, say 60 degrees (this means BTotal will be set to 60), we'll tell Isotope to extend its actuator until that 60 reaches a number less than five, while subtracting three from it every time the My Block is run. In this case, it will take just under two seconds for Isotope to stop moving his actuator (1.9 seconds = 19 runs of the My Block; 19*3 = 57; 60-57=3 and three is less than five, which is our threshold for movement).

I hope this makes sense! [Okay, I realize it probably doesn't. Mostly I hope I don't get any angry emails about it!]

Anyhow, thanks for reading The NXT Big Thing! See you NXT time! **SV**

Author's Note

Hey all! Many of you know I run a small business called Techsplosion bringing hands-on science (LEGO Robotics included!) to kids of all ages. Well, business has been booming lately, and in a bittersweet discussion with Robin, the fantastic Associate Publisher for *SERVO* and *Nuts & Volts*, I've decided to switch to publishing The NXT Big Thing every other month, so I have more time to dedicate both to Techsplosion and to The NXT Big Thing (without freaking out and missing deadlines. Huzzah!) So, we'll still be around, but we'll switch to an every other month publishing schedule. In the meantime, if you're a Bay Area resident and want to get your LEGO fix, check us out at **Techsplosion.org**! *Greg*

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Then and NOW

Eddie: RDS4 Meets Kinect Meets Parallax

by Tom Carroll

Last month, I discussed *Willow Garage's* TurtleBot — an advanced experimenter's robot based on WG's ROS (Robot Operating System) which was first developed at Stanford. This unique robot uses the iRobot Create as the motive base for the robot. The key component is the Microsoft Kinect sensor system developed for the Xbox 360 game system. The TurtleBot follows in the footsteps of its big brother — Willow Garage's $400,000 PR-2 personal robot designed for serious (and deep pocketed) robot experimenters.

This month, I want to cover another advanced experimenter's robot that seems to be as impactful to the robotics market as the TurtleBot. This new machine is Eddie, shown in **Figure 1**. It was developed by Parallax in conjunction with Microsoft (the developers of RDS4, Robotics Developer Studio) and the Kinect sensor. I'll be comparing features of both robots at various points in this article. As with the TurtleBot, Eddie uses a laptop computer for the main processing power. However, the laptop for Eddie is left opened and the keyboard is available to use on the robot, whereas the TurtleBot has its Asus netbook closed and inserted in one of the slots. Eddie is certainly not the first robot to use a laptop; recall amateur builder Dave Shinsel's Loki that uses an 'exposed' laptop for a processor as shown in **Figure 2**.

FIGURE 1. Parallax's Eddie robot.

FIGURE 2. Dave Shinsel's Loki.

Evolution Robotics ER-1

The use of an exposed laptop atop a mobile robot was also the design of a commercial robot by Evolution Robotics. Evolution's $599 ER-1 personal robot is shown in **Figure 3** and was introduced about a decade ago. It operated on what they called 'RDK,' or their Linux-based Robot Development Kit. The ER-1 was also available as a kit for $499 and used an 'x-beams aluminum building system,'

with a pair of stepper motors and a motor control board, a USB camera, and the GUI RDK software package. The ER-1 was used in many schools as a teaching platform. Evolution is now better known for their Mint floor cleaning robot shown in **Figure 4**, and their NorthStar navigation system that it utilizes to maneuver around a home's floors in straight lines.

The open laptop configuration and special software package is what makes Eddie stand out from its competition. Since this robot is not intended to be a fancy robot for the masses — but rather a robotic platform for serious experimentation — I personally feel that the exposed laptop configuration is preferable to a hidden computer. (However, Willow Garage's TurtleBot with the hidden netbook seemed to work just fine.)

FIGURE 4. The Mint floor-cleaning robot.

FIGURE 3. Evolution Robotics' ER-1.

The Parallax Line of Robots

Parallax has long been in the forefront of experimental robotics. Their Boe-Bot kits have been used by experimenters and students of all levels to learn about robot construction and programming. The $170 Boe-Bot is based on the Parallax Board of Education (thus BOE roBOT). The $70 Board of Education is a BASIC Stamp development board with a solderless 1-3/8" by 2" breadboard area; it is used in many Parallax kits. Their $49 BASIC Stamp 2 module contains a PIC16CC57c microcontroller from Microchip Technology. There are several levels of BASIC Stamps that use a variety of microcontrollers applicable to the designer's varied requirements, but most people find it more cost-effective to purchase a complete robot kit, such as the Boe-Bot shown in **Figure 5**. It just costs slightly more than the separate microcontroller and the BOE.

The recently-announced PropBOE development board — a prototype of which is shown in **Figure 6** — is based on Parallax's eight-core Propeller multitasking microcontroller. Gordon McComb wrote an excellent article in the December *SERVO* about a beginner's robot using this very capable chip/board. The board is scheduled for production in early 2012 after some final tweaking is done.

The very popular Boe-Bot is not the only robot kit from Parallax. There is also the Scribbler 2 shown in **Figure 7** which costs $130-$140, depending on a serial or USB configuration. This robot got its name from its ability to use a Sharpie pen to draw programmed figures. It is a nice programmable robot with several built-in demos.

The $149 Stingray robot chassis is a quality starting base for your own design. The kit shown in **Figure 8** is made of anodized aluminum alloy. Parallax also has two SumoBot kits ranging from $160 for a single robot to $240 for the complete Competition kit with two robots as shown in **Figure 9**.

The QuadRover shown in **Figure 10** is now actually on sale at $3,500 because it will soon be discontinued. This 89 pound robot is powered by a gas engine driving a hydraulic system and is controlled by Parallax's eight-core Propeller processor; it can be remotely controlled as delivered. The

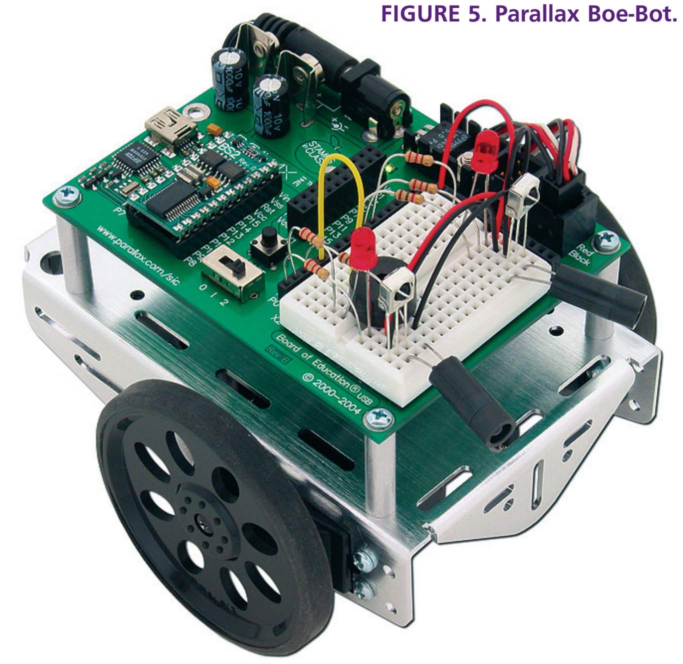


FIGURE 5. Parallax Boe-Bot.

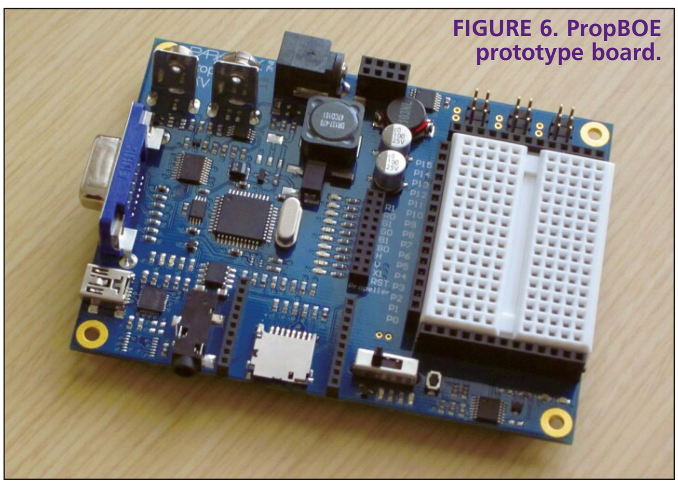


FIGURE 6. PropBOE prototype board.

FIGURE 7. Scribbler 2 robot.

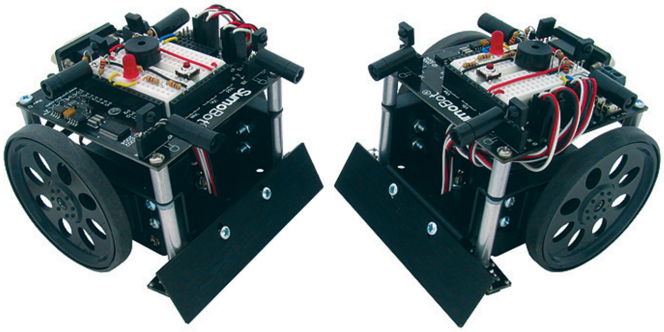
user can add GPS, a compass, and accelerometers for autonomous capabilities.

The Parallax Eddie

The Eddie robot is the new kid on the block for Parallax and is fairly powerful. According to Microsoft, RDS4 includes a simulation of — and full support for — an affordable and capable hardware reference platform known as Eddie (Expandable Development Disks for Innovation and Experimentation). It is available only as a kit and is priced at $1,249. Created by Parallax, this device is a robot that can roam autonomously, see in 3D using Kinect, and be driven remotely using a wireless controller. As with TurtleBot, Eddie was developed around Kinect's camera sensor which was developed for the Xbox 360 and is a turnkey robot system that allows advanced experimenters to develop a sophisticated robot using their own laptop computer for processing power and Kinect.

Figure 11 is from the **quora.com** website and depicts how Kinect works for both robots. (You can read more about Kinect in last month's TurtleBot article, or go on-line for hundreds of hacker sites and thousands of articles.) Unlike TurtleBot, Eddie uses RDS4.

FIGURE 9. A pair of Parallax Sumo robots.



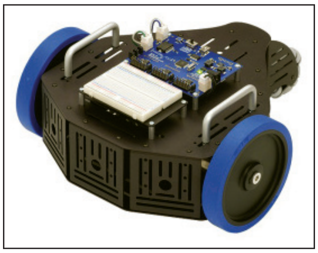


FIGURE 8. Stingray robot.

You might wonder just how Parallax and Microsoft got Kinected. (Sorry for the pun.) I spoke with Ken Gracey of Parallax about the relationship between the two companies. "Around June 2011, they <Microsoft> contacted us about being the Robot Developer Studio reference design," Gracey explained. "They had purchased many robots and components from a number of suppliers for evaluation. Because of their software schedule and legal considerations, they were giving their chosen supplier — Parallax — only a few months to produce the robot that they wanted."

"They told us that they chose Parallax because we have our own in-house manufacturing (CNC mills, routers, laser cutters, P&P machines) and because we produce good quality robot parts. To my knowledge, there were no other suppliers under consideration. We agreed to meet their request and proceeded with a design. First, we had to give them the hardware with hand made PCBs — wire-wrapped and soldered. Then, we produced the Eddie Control Board, along with a second revision to fix some power supply issues. About 90 days into the effort, we were able to provide exactly what they wanted," Gracey continued.

"At this point, Parallax is the key reference platform. Other companies can produce an RDS-capable robot as long as they use the protocol we produced with Microsoft. It could be any microcontroller, but the Propeller really shines in this application because of the multicore design. Parallax had worked with Microsoft in the past on the MSRS system with our Boe-Bot, so we also had some experience working within their system. It's been a rewarding process," he concluded.

In response to Microsoft's needs for a robotics platform and Parallax's own future product line growth, Gracey

FIGURE 10. Parallax QuadRover.

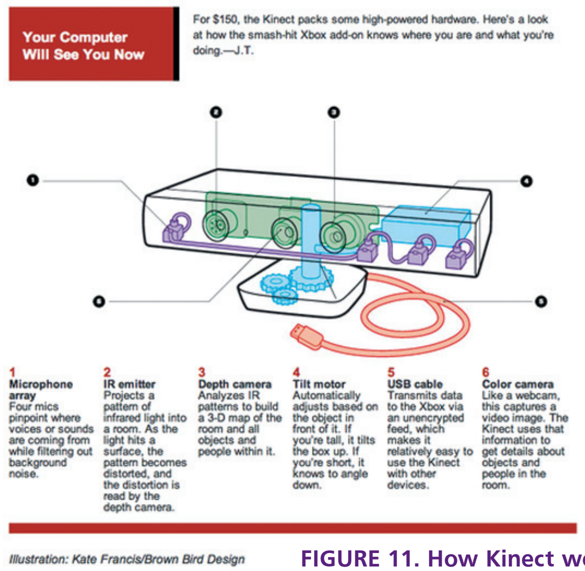


FIGURE 11. How Kinect works from Quora.com.

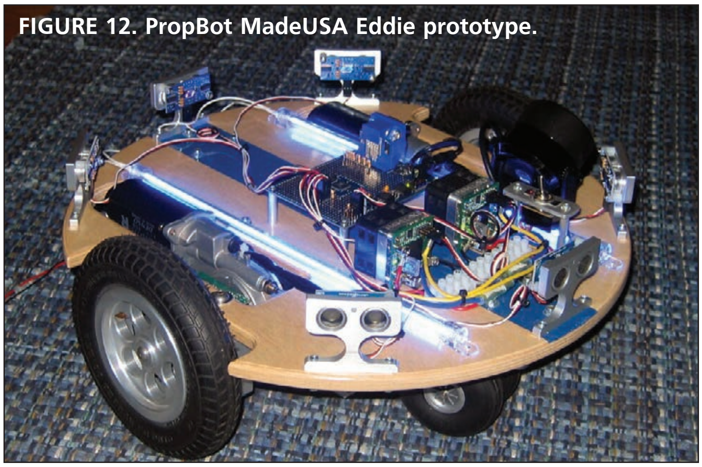


FIGURE 12. PropBot MadeUSA Eddie prototype.

FIGURE 13. TurtleBot and Eddie.

personally worked on the development of a platform that would eventually be the MadeUSA (discussed later) and Eddie base. **Figure 12** shows one he first developed on his own in his garage using high grade plywood for the base and machined aluminum Ping))) ultrasonic sensor holders, with fluorescent highlight lighting to add a bit of 'bling.' Microsoft had a roborazzi (ROBOt papaRAZZI) robot using this base that roamed parties, taking photos of the individuals.

As the idea for a Kinect-based robot was bounced around the Parallax offices and at Microsoft, Gracey handed the leadership of the Eddie project over to Jessica Uelmen, Engineering Manager. I got a lot of info from Jessica about the fine points of the system, as well as the installation of the many parts of the RDS-based and Parallax software.

Eddie and TurtleBot Comparison

Figure 13 shows Eddie and TurtleBot side by side. You can immediately see the major physical differences in the two machines. TurtleBot is 18 inches tall (without any top-mounted experiments) by 12-1/2 inches in diameter. Eddie is 24.5 inches tall (to the top of the Kinect sensor) and 17.8 inches in diameter. It weighs quite a bit more at 25.3 pounds without the computer and Kinect, and a bit over 30 pounds with a laptop and Kinect. Eddie is capable of handling a payload of up to 50 pounds which means you could attach two large articulated arms on each side and a slew of sensors.

One of the main differences in the two robots is that you furnish your own laptop and Kinect sensor for Eddie; the TurtleBot comes with an Asus netbook and Kinect. Many experimenters prefer to use their own computer and software systems (they also might already have Kinect that is used for their Xbox 360).

Software and Computers

The main computing power for the two robots is pretty much the same since they both use a laptop/netbook for this purpose and the Kinect sensor. Willow Garage chose mid-structure mounting for the Kinect sensor on the TurtleBot, thus allowing the addition of experiments (such as an arm mounted on the top plate). The Eddie design chose the open-laptop configuration for access to the computer's keyboard with the Kinect sensor mounted above the computer. This allows for a better view for the Kinect, but limits the addition of mounting any sort of experiments at the top.

The other main distinction is the robot's software. It's mentioned, Eddie uses the RDS4 Beta to control the Kinect sensor, the other five sensors, and motion control. TurtleBot uses the Stanford University developed Robot Operating System or ROS to control the Kinect and the iRobot Create base. ROS is open-source — one of the main reasons Willow Garage developed the TurtleBot. Microsoft is a little more careful about that term and the use of RDS4, though robot experimenters can download the software from their site for free.

Mechanical Construction

If there is a clear mechanical distinction between Eddie

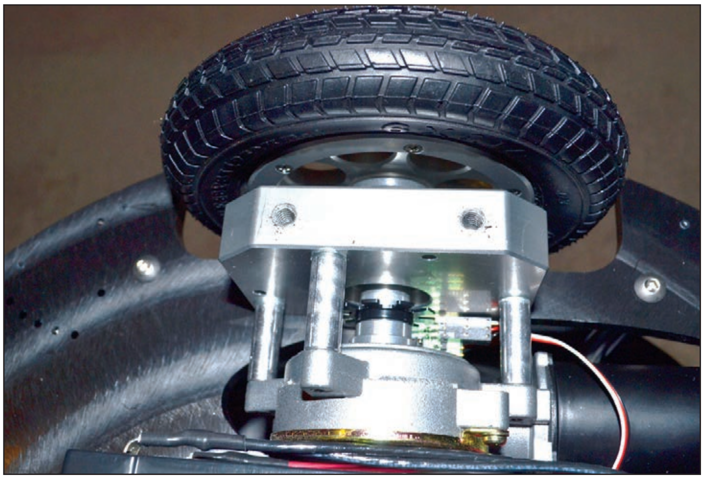


FIGURE 14. Wheel mounting and encoder.

and the TurtleBot, it is the construction and robustness of the robot's motive base. TurtleBot uses the iRobot Create as its base since it is packed with sensors and is basically ready to go when its battery is charged. Parallax chose to develop a sturdy base of its own design.

The base consists of two 18" diameter mounting discs that are made of milled, high density polyethylene (HDPE) on which the two ultrasonic and three IR distance sensors, the controller board, and the two drive motors are mounted. A bottom plate holds the two gelled–electrolyte 8 AHr 12V batteries to keep the center of gravity low for stability.

The Eddie kit will take the average person approximately 4-6 hours to assemble. Eddie seems to be simple to construct, though there are many parts in the kit. The instructions for Eddie consist of 16 pages that are very well written, with color photographs instead of line drawings. The quality of the robot is evident from the two very quiet worm-drive motors used on the main wheels.

Typical with all of Parallax's higher-end robots, the sensor and motor mounts as well as the main wheel components are machined 6061-T6 aluminum for the wheels, gearmotor mounts, caster swivels, and HDPE for the sensor mounts. Parallax built into their wheel/drive motor components what most robot designers fail to do with large robot construction: They added an outboard bearing to compensate for the wheel axle bending moment. This allows for a greater payload without bending the gearmotor's output shaft.

Figure 14 shows one of the machined wheel mountings and the optical interrupter shaft encoder. There are even two holes on each side for lifting as the robot is a bit heavy. The edges of the holes are a bit sharp and uncomfortable to grip for me, but this can be remedied by placing two short pieces of foam pipe insulation over the holes.

FIGURE 15. Robot's dashboard with Jessica Uelmen.

Electronic Components

The most visible electronic components are the five sensors mounted around the front of the robot. These sensors include three Sharp 2YOA21 infrared distance sensors with a 10 to 80 cm range, and two Parallax Ping))) 28015 ultrasonic sensors with a 2 to 300 cm range to detect objects to help avoid collisions — this provides sight where the Kinect cannot see. Eddie can navigate and interact within dynamic environments using sensor fusion to integrate the Kinect's 3D vision, color imaging, and sound processing with the platform's wheel encoders. Eddie is also expandable and hackable, making it easy to add sensors, accessories, and custom add-ons.

The heart of Eddie's control board is the Propeller P8X32A microcontroller. The Propeller chip's eight 32-bit cores provide incredible power and flexibility. The board includes high current motor drivers, an eight-channel 10-bit ADC, and access to lots of digital I/O. Multiple regulated power supplies (12V, 5V, and 3.3V) and three switchable auxiliary power ports support optional accessories such as an under-body LED lighting display (or similar). Eddie is powered with a pair of long-lasting dual 12V, 8.0 Ah gel-cell batteries. A 3A smart battery charger is included. Fully charged batteries provide a typical run-time of around 4–7 hours, so you can keep innovating without interruption.

As you might remember from my article on TurtleBot last month, the Kinect was originally developed for the Xbox 360 game console as a way for users to play games through arm and body movements as well as spoken words, without using a hand controller. The Kinect sensor has an infrared depth sensor, an RGB camera, and a microphone array as input devices for game (and robot) control.

Seeing tremendous non-gamer interest in February of last year, Microsoft announced that it would release a non-commercial Kinect for Windows Software Development Kit (SDK) in June. Unlike the TurtleBot with its Linux-based ROS, those with a C++ or Visual Basic background should have no trouble programming in the Windows environment.

Software Installation

Software installation requires a few more steps than

the following; I've shortened some of the smaller steps for this article. Before installing 'Eddie' on the robot, there is an order of programs you'll need to install first:

Go to **www.microsoft.com/visualstudio/en-us/try** and install Microsoft Visual Studio 2010 and Windows SDK. There is a 60-day free trial available here (any version is fine). You'll also need:

- SpeechPlatformRuntime_32bit
- MSKinectSpeechPlatformSDK_32bit
- MSKinectLangPack_enUS
- KinectSDK-v1.0.-beta2-x64

At this point, you're ready to install Microsoft Robotics Developer Studio 4 Beta 2 (currently running RDS is very software intensive). After doing so, open the ParallaxKitReadMe for instructions on installing the Eddie services. All downloads and installation instructions and information are available at **www.parallax.com/eddie**.

Figure 15 shows the robot's dashboard — a means of controlling the basic movements of the Eddie platform by driving the two main wheels. The dashboard displays distance data from the IR and ultrasonic sensors. Eddie can also be driven remotely by the Xbox 360 wireless controller. Uelmen is shown in two camera views of the Kinect sensor — the depth camera's 'silhouette' view and a color view from the RGB camera.

Figure 16 shows a view of the depth camera data with a skeletal overview. The black splotches on the screenshot depict areas out of the Kinect's field of view. Kinect can be queried by checking the required boxes in the upper left corner. This Kinect user Interface is available from the RDS4 package **Figure 17** shows the skeletal overview from the RGB camera's full color view, showing all three data points checked.

It is strongly recommended that you place some books or blocks under Eddie's battery mount plate to lift the drive wheels off the floor or workbench when first powering the robot up and installing the software. As with any experimental robot, they can suddenly come to life in a way you're not expecting, and either dart off and run into a wall or drop to the floor with disastrous consequences. (I had one do that to me when I had just finished it.)

Develop Your Own Eddie Robot

Readers who are interested in building an Eddie-style robot may want to consider deviating from the Parallax design and creating their own configuration. Without having to cut new metal yourself, Parallax has another kit that just might meet your needs: the $880 MadeUSA shown in **Figure 18**. You might wonder where Parallax got that name. Well, check out the Jan '12 article by Gordon McComb for a detailed explanation.

This base is identical to Eddie in most respects but is a single-plate base with a pair of Parallax-made HB-25 fan-cooled electronic speed controls and 10 of their Ping))) ultrasonic range finders (no Sharp IR distance rangers). Batteries are not included but similar batteries to Eddie's can be mounted under the main plate.

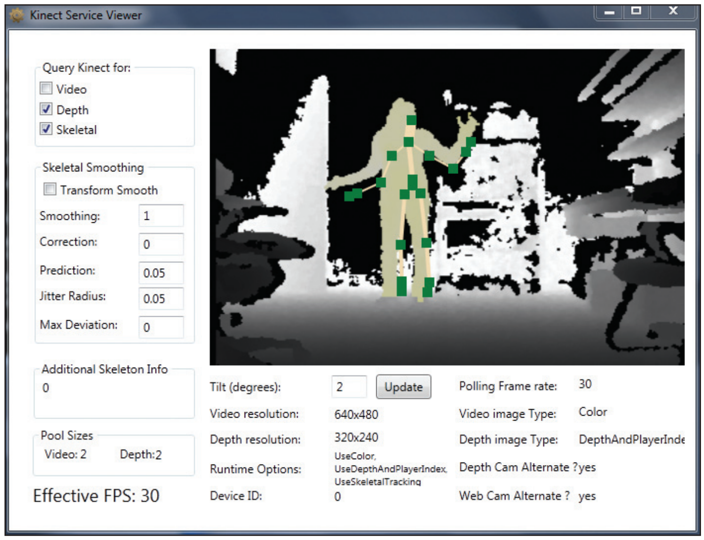


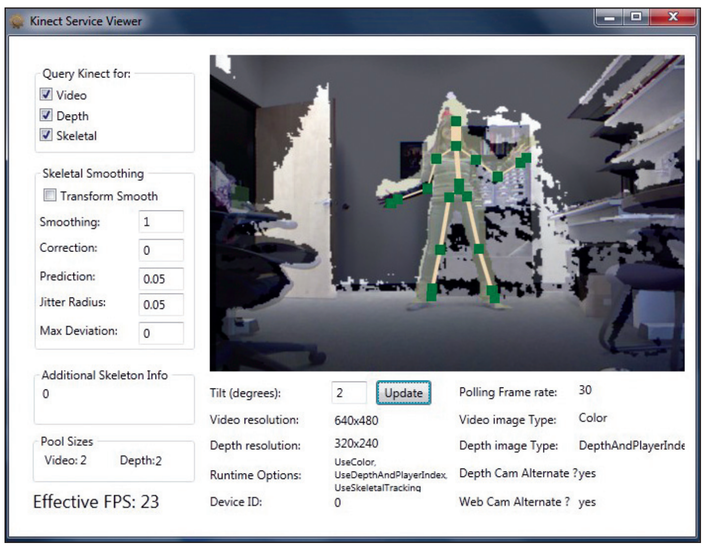
FIGURE 16. Kinect depth and skeletal views.

Builders can use their own processor and develop their own code or download code from a number of sources. The same quality gearmotors, motor mounts, pneumatic wheels, and casters are included. The machining for the rims, mounts, swivels and Ping))) mounts is from aerospace grade 6061-T6 aluminum. Eddie's control board is not presently available as a separate product but experimenters can still use their own designs and microcontrollers to produce the PWM output for the gearmotor's ESC requirements, as well as processing data from the ultrasonic range finders and the complex Kinect data.

Final Thoughts

I've reviewed and experimented with two incredible machines the past two months. Each one can perform

FIGURE 17. Kinect depth, skeletal, and video overlays.



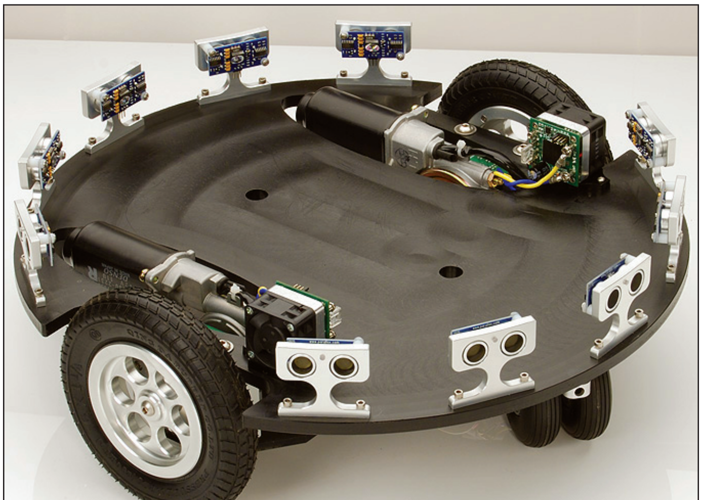


FIGURE 18. MadeUSA robot base kit.

routines that would have been impossible for robots under $15,000 just two years ago, yet these machines cost a fraction of that and are absolutely amazing. Eddie almost feels alive to me as does TurtleBot. They both have the capacity to speak and to understand speech which is now so easily accomplished with today's robots.

What's next? Covering these robots with artificial skin, giving them a face, and sending them out on the streets, perhaps.

Should you buy one or build your own? Is Eddie the best choice or should it be the TurtleBot? I personally like Eddie — mainly for its size and payload capacity, but that is my choice. Everyone has their own requirements and needs. Since both machines have the Kinect sensor system, the main difference is really the software: ROS or RDS4. Again, I feel more comfortable with RDS4, but again, that is my choice.

Both platforms have certain superior qualities over the other as well as inferior points. Eddie is more expensive but seems to be worth it for its quality and robust construction. TurtleBot is not nearly as upgradeable for heavier add-ons since it is quite a bit smaller. TurtleBot comes with the Kinect and a netbook whereas you must furnish your own for Eddie — which could actually be a plus for many experimenters. Do take the time to look at these machines if you are at least a tiny bit interested in expanding your personal robotics knowledge and capabilities by a large magnitude. These machines add a new chapter to personal robotics experimentation. SV

Tom Carroll can be reached at TWCarroll@aol.com.

Advertiser Index


DF RobotShop
Rover
Compatible with
ARDUINO
MYROBOTS.COM
Connect your Robots. Reap the benefits™
DF RobotShop Rover
www.RobotShop.com
DFROBOT
www.DFRobot.com
Affordable, Flexible and Scalable
Compatible with thousands of off-the-shelf parts!
XBee
Wireless Control
Speech Control*
Mecanum
RobotShop
Robotics at your service!™
www.robotshop.com
Scan on your smart-phone
with a bar reader app
ADDITIONAL VARIANTS TO COME
THE EVOLUTION CONTINUES...
RobotShop and MyRobots logos copyright RobotShop Distribution inc. Robotics at your service and Connect your Robots. Reap the Benefits are trademarks of RobotShop Distribution inc. DFRobot logo copyright DFrobot.
Arduino logo copyright Arduino. Xbee logo copyright Digi International Inc. *Headset and female operator not included with speech controlled model, which is provided with an integrated electret microphone.

Prices and availability subject to change without notice.


What's New?
DARwIn-OP consists of
20 * DYNAMIXEL MX-28,
the same model for all joints.
For more info, please visit
www.RobotSource.org
DYNAMIXEL
MX-Series
MX-28
MX-64
MX-106
S/W Tools
MATLAB®
LabVIEW
Microsoft Visual Studio
python
C/C++ C# Visual Basic
Java
Contactless Absolute Encoder
12-bit resolution, 360° operating range with super-durability
P I D control
User Programmable PID Control Gain
Precise and reliable PID control
360° 4096 steps
Powerful Resource & Organized Design
Maxon motor, 32-bit controller,
high communication speed
RX and EX series compatible dimension
Torque
Current Based Torque Control
Position and speed control
with dual-loop current control
(MX-64, MX-106 only)
Usage
[USA] E-mail : america@robotis.com Tel : +1-949-333-3635
[KOREA] E-mail : korea@robotis.com Tel : +82-70-8671-2604
[JAPAN] E-mail : japan@robotis.com Tel : +81-3-4330-3660
[Other Countries] E-mail : contactus2@robotis.com Tel : +82-70-8671-2609
ROBOTIS
www.robotis.com
© OLLO, BIOLOID, and DYNAMIXEL are registered trademarks of ROBOTIS CO., LTD.